

Smart Eco System - Future of Gardening

Dorota Marczak, Scott Allan, Nader Sobhi

CMP311: Professional Project Development & Delivery

2018/2019



Please note that Information contained in this document is for educational purposes.

Abstract

Background to the paper (brief and client) and aim of project

With the advent of new technologies, which have revolutionized countless areas of human life, many employers realized that multiple activities may now be automated, or even performed remotely at relatively low costs. Amid trending tech solutions such as 3D printing and wireless conference rooms there is one inconspicuous, yet very powerful tool - the Internet of Things. IoT is a concept which emerged barely a decade ago and has turned out to be a blessing especially for small companies, which constantly compete for shares on the market while having a limited budget and human resources. IoT technology can be used for a variety of simple activities performed with users consent and according to their instructions such as Amazon's Alexa smart speaker, a staple in many homes today.

Dr Lynsay Shepherd, representing Abertay Plant Systems, decided to take advantage of the opportunities created and ask the local tech start-up - DNS Team – to design and develop a plant monitoring system utilizing the above-mentioned technology, as well as similar web-based solutions. The client stressed that the company expects a device that could be used by anyone; not only to appeal to their existing customer base but also to “tech obsessed Millennials who care about the environment but don't keep any plants in their home”, as Dr. Shepherd put it. The idea to merge contemporary technology with the age-old art of plant keeping was perceived by DNS Team as a unique challenge.

The aim of the project, settled on after closer familiarization with the provided brief, was to provide an easy to use system that required no expert knowledge to operate and maintain. Ideally, a ‘plug and play’ device would be developed that would begin operation as soon as the end user registered the device on the companion website. Once registered, the device would start to collect data which would then be displayed via the website, as well as via a built in LCD screen.

Methodology

In order to deliver a high-quality product within the agreed timescale, development of the hardware system and the website was conducted concurrently, which minimized the risk of a potential delay. Team members were assigned to certain tasks according to their strengths, which facilitated breaking the work down using the Sprint methodology. Following the Scrum methodology allowed the team to set clear goals over the course of the development phase, meaning the team leader could track the project progress over time. Regular meetings (see *minutes*) were held to ensure the project proceeded as planned, the workload was distributed fairly and finally, that all team members have a chance to air any problems they may have faced with a certain task.

The final product consists of three main elements: Arduino board with connected peripherals, a database which stores the sensor readings, as well as the website that provides an overview of the well-being of the plant. The physical device processes the data received from the attached sensors and sends them in JSON format to the database via Wi-Fi connection, provided by the WEMOS adapter. The data is held in normalized form and then accessed by the website, that utilizes it to create intuitive data visualizations.

Outcome

The professional attitude of each team member to their entrusted tasks resulted in delivering high-quality product a few weeks before settled deadline, what may turn out to be highly beneficial for client's business and its growth. Smart Eco System, a user-friendly and visually attractive interface providing reliable information about a plants condition, has a chance to become a bestseller in the gardening market, which has not fully experienced a technological revolution yet. This fact, in turn,

may attract the attention of gardeners devoted to their plants and therefore create a model for smart gardening devices.

Contents

1	Introduction	5
1.1	Background	5
1.2	Aim	5
2	Procedure.....	7
2.1	Overview of Procedure.....	7
2.2	Procedure part 1	7
3	Results.....	20
3.1	Results for part 1.....	20
4	Discussion.....	32
4.1	General Discussion	32
4.2	Conclusions.....	33
4.3	Future Work.....	33
4.4	Call to action	34
	References	34
	Appendices.....	Błąd! Nie zdefiniowano zakładki.
	Appendix A - Example.....	35
	Appendix B – Deliverables and Requirements	Błąd! Nie zdefiniowano zakładki. 43
	Appendix C - Minutes	46
	Appendix D – User manual.....	Błąd! Nie zdefiniowano zakładki. 56

1 INTRODUCTION

1.1 BACKGROUND

While it is true that you can happily grow your plants with a packet of seeds, a sunny patch of land or a simple flower pot filled with soil and a watering can, there are much better and faster ways to create your dream garden. Less than a decade ago, modern technology crashed into the gardening sector that was experiencing developmental stagnation, interrupted from time to time by new biochemical solutions. Despite almost unlimited opportunities brought by technological revolutions, many gardeners were still confining themselves to obsolete cultivation techniques, originating from the first half of the twentieth century and some even from ancient times. In many cases, the reason behind it is not related to the potential costs, but rather the habit of following traditions and not departing from the things we already know. However, with the advent of GMOs, people began to realise that it is possible to take control over their plants fate which, until recently, remained almost entirely dependent on weather conditions. That, in turn, switched on the green light for a whole range of modern solutions, including intelligent automatic watering and remote growth optimization based on cause and effect analysis. It must be noted, that it was Millennials with their concept of a need to be the mother of invention that have led the revolution to success. Instead of relying on a gut feeling and mercurial weather conditions, they prefer to influence nature by utilizing in-depth exploration of a plants environmental preferences and delivering the required mineral resources based on extensive analysis.

According to the Washington Post, Millennials began paying attention to the plants they place in their urban flats that not only serve as a live decoration or a reference to a favourite aesthetic pattern but are also a way of expressing the unique personality of their owner. As ubiquitous social media encourages the younger generation to share every aspect of their life online, modern society is under pressure to be perfect in every aspect of contemporary life. However, the first steps in gardening often turn out to be harder than expected. They browse the Internet to find the tips for geeks, but then forget to water their plants regularly and end up trying their luck again. Finally, they reach for gardening applications as a last resort in their gardening adventure. Unfortunately, instead of fully-fledged systems they find crawling applications with basic advice and watering reminders being the extent of their possibilities. Having no choice, they install one of them on their phone and begin anew. This time they do not forget about basic needs of their plants and that is usually enough to keep them alive for a few weeks. Proud of themselves, they enlarge their little collection with a few more specimens, which turn out to be even more demanding. Irritated by imprecise advice available on potted plant labels or seed packaging, such as 'place in partial shade' or 'requires medium temperature environment', as well as hard to estimate irrigation needs, they look for a tool that would facilitate taking care of their flowers. They are eager to pay for a solution to their problem, but unfortunately none of the available products fulfil their needs, either due to being too expensive or requiring specialised knowledge to construct.

1.2 AIM

The fundamental high-level aim of the Smart Eco System project was creating a product that would meet the needs of modern gardeners; both digitalized Millennials just discovering the world of plants, as well as older members of society experienced in the field, but wary of new technologies. In practice, the project aim is to deliver a device that would collect environmental data on a plant and place the

readings into a database. This data would then be used by a web interface to power intuitive and visually-attractive graphs. The complete system itself, being the outcome of the project, will give the user the means to monitor their plants as precisely and meticulously as if they were placed in a scientific laboratory. For an experienced gardener, such an opportunity could become the beginning of plant growth optimization, which could potentially save them time and money.

Although the main goal of the project was to design and build the Smart Eco System along with a user interface, there were several other objectives identified. Firstly, it is expected that the system will provide a set of features, such as an accessible and responsive website, that will adapt to different screen sizes. The system should be easy to maintain, as basic training provided to the members of staff will not cover the full technical specification and the staff themselves may not have technical expertise. Furthermore, it is crucial that the application is user-friendly and visually attractive, otherwise it may dissuade potential customers from purchasing the product. Another important factor is safety – malicious attacks may cause interruptions in service and these, in turn, may lead to claims of a lesser product if they occur frequently. By treating sensitive data with particular care, the company protects itself against credential leaks, which in turn ensures that they will not have to pay out of pocket to compensate for any damages they may have caused. This would also prevent the tarnishing of the image of a reliable brand.

Regarding the physical device, a vital feature must be reliability combined with a smooth flow of data, as the company cannot afford releasing a poor-quality product. Reducing possible data loss is a minimum requirement. Last but not least, collected data must be stored in a manner this is compliant with the relevant laws in operation and its integrity should be preserved throughout the system's life-cycle.

2 PROCEDURE

2.1 OVERVIEW OF PROCEDURE

According to Duncan Haughey, author of Project Smart platform, the key to a successful project is detailed planning. That is also why the next step, taken after setting the goals of the project and identification of the stakeholders, was dividing the development stage into sections corresponding to the deliverables. Each of them consisted of a number of features, which had been established based on an extensive list of functional and non-functional requirements. Before any development work began, the organization of the database had to be completed in a way that would enable reception of data from the physical device, as well as to allow the website to read and process the data. The team then proceeded to the prototyping stage, where each part of the system was analysed in terms of both its internal operations, as well as cooperating with other elements in the set. Once the prototypes were up to the desired standards the respective procedures of each of the system components began.

2.2 PROCEDURE PART 1 – ARDUINO

The hardware device was developed incrementally to ensure that each stage was successful and to allow time for appropriate testing and modification as required. By building the device in small stages, it allowed the team to track progress as well as served as a troubleshooting guide should an error occur. For example, if the device was behaving normally before a sensor was added, only to misbehave upon integration, it would seem the cause of the error lay with the incorrect installation of the new sensor.

This section will look at how each stage was carried out and how the overall solution was reached. Connection diagrams will be provided for each stage, with a complete diagram incorporating the whole hardware system included at the end of the section.

Connect Arduino and WEMOS

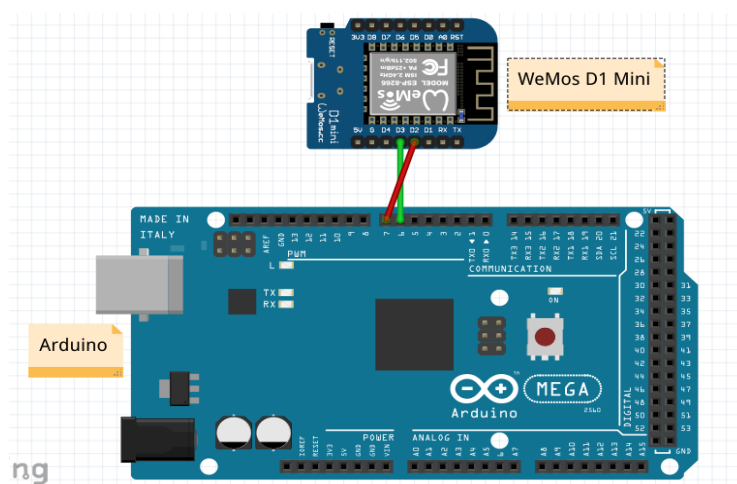


Figure 1

Before we could gather the required data from the sensors to send to the database, we had to give the Arduino the capability to connect to the Internet. This was achieved using a WEMOS D1 Mini board. The WEMOS is powered by a micro USB connection and uses two pins to send and receive

data to and from the Arduino, making it relatively simple to connect and setup. The code running on the WEMOS board consists of establishing a connection with the database, listening for the data collected by the sensors via the Arduino, placing the sensor values within a JSON string and finally inserting the sensor values in the appropriate table. Once a connection with the database has been established, it will construct and send a JSON string every thirty minutes. Before any sensors were connected, placeholder data was used to ensure that a WiFi connection could be set up between the device and the database server, as well as to ensure that the data was being stored in the database correctly. Once this stage was complete, the next step was to connect a sensor and obtain useful data.

Connect Temperature / Humidity Sensor

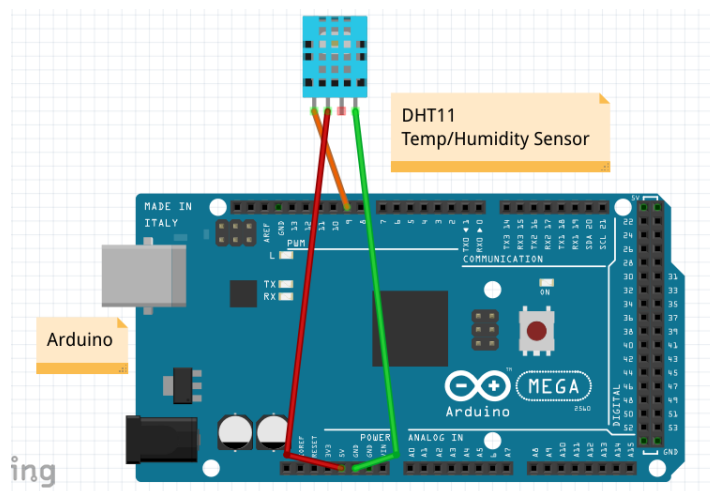


Figure 2

The DHT11 sensor was selected as it enabled the team to take readings on the temperature and humidity in one unit. This sensor requires three pins to operate. It is powered via a 5V pin, requires use of a ground (GND) pin and transfers data via a digital pin. A DHT library is also required for successful operation and enables the use of functions specific to the DHT11. The `dht.begin()` function starts up the sensor, whilst the `dht.readHumidity()` and `dht.readTemperature()` functions translate the data received from the sensor into a human readable format. These values are read as floats before being converted to String objects for use with the JSON string.

Connect Light Sensor

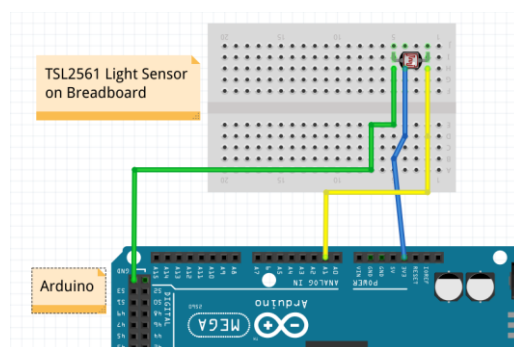


Figure 3

The TSL2561 was chosen for use as the light sensor as it is very simple and provides easy to use data. The sensor reads the amount of light falling on it in a given environment and outputs this data on a scale from 0-1000. This allowed the team to save the data values as an integer and display it to the

user as a percentage with 0% meaning fully dark and 100% fully bright. The sensor requires three pins to operate: 3.3V to supply power, GND to ground the circuit and an analogue pin to provide the data, in this case the A1 pin. A breadboard had to be used as the sensor is very basic and is not attached to a board to provide the pin out functionality.

Connect Moisture Sensor

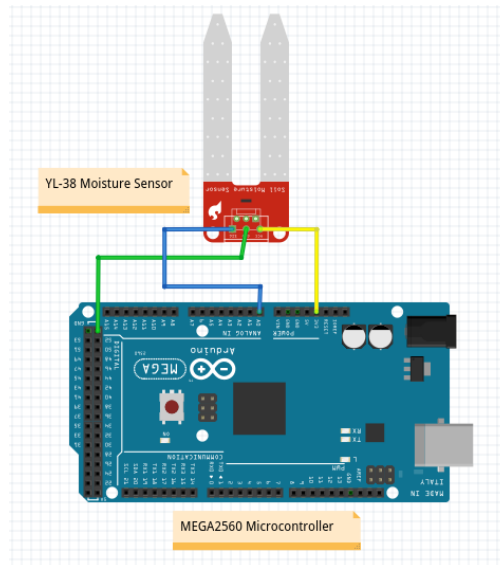


Figure 4

The YL-38 moisture sensor was chosen as it is simple to connect to the system and provides data that is easy to interpret. The sensor is powered from a 3.3V pin, grounded via a GND pin and sends data through an analogue pin, in this instance A0. The sensor works by reading the amount of electrical resistance in the circuit. The two prongs are connected to the power pin and the input pin and the difference in the returned value is used to calculate how much moisture is present in the given medium. Should the sensor be placed in wet soil, a connection between the two prongs will be established and the value returned will be lower than if the sensor is placed in dry soil where the current finds it harder to travel through the medium. Using this data, we can inform the user whether their plant requires watering or not by using a simple if-else statement to determine where the returned value falls on a scale.

Take Readings & Place in JSON String

Once all the sensors were connected and providing useful data to the Arduino/WEMOS sub-system, the next step was to place these data readings in a database. There was a number of options the team could have chosen, but the decided method was to insert all the data values into a JSON string and store this string within a database. The advantage of this method was that the database table only required four columns: one which held the unique identifier, in this case the MAC address of the WEMOS, a column to hold a timestamp of the reading, and a column which contained the JSON string itself. The fourth column was an autoincrementing integer required for SQL operations. The team could then parse this JSON string using PHP and use the data to power the graphs on the website. The disadvantage of this method is that the JSON string won't be sent to the database until it contains four separate readings [one for each sensor], meaning that database entries could be missed if a sensor became inoperative.

```

if (mvalue1>=6 && tvalue1>=6 && lvalue1>=6 && hvalue1>=6){
    if (mresult==" " && hresult==" " && tresult==" " && lresult==" "){
        digitalWrite(11, LOW);
    }
    jsonBuffer["m"]=mresult;
    jsonBuffer["h"]=hresult;
    jsonBuffer["t"]=tresult;
    jsonBuffer["l"]=lresult;

    serializeJson(jsonBuffer, Serial);
    String jsonstr="";
    serializeJson(jsonBuffer, jsonstr);
    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);
    String q = "INSERT INTO DDNS_DB.plant_readings(device, time, readings) VALUES ('"+WiFi.macAddress()+"',NOW(),'"+jsonstr+"')";
    char query[512];
    q.toCharArray(query, 512);
    cur_mem->execute(query);
    jsonBuffer.clear();
    mvalue1=0;
    hvalue1=0;
    tvalue1=0;
    lvalue1=0;
    delay(1800000); //30 minute delay sending sensor data to dB
}

```

Figure 5 - Send json string to dB

The WEMOS is used to construct the JSON string before transferring it over WiFi to the database. The Arduino collects all the sensor data and passes it onto the WEMOS via a digital pin. The WEMOS listens for the sensor values and once it has collected all four required readings, a JSON string is created with the values being appended in the appropriate location. The WEMOS then constructs a MySQL query consisting of the MAC address of the WEMOS which is used as a unique identifier, the current time and the JSON string itself. Once the MySQL query has been executed the JSON buffer is cleared so that it can be used to construct the next JSON string. A delay of thirty minutes is set after which the process begins anew.

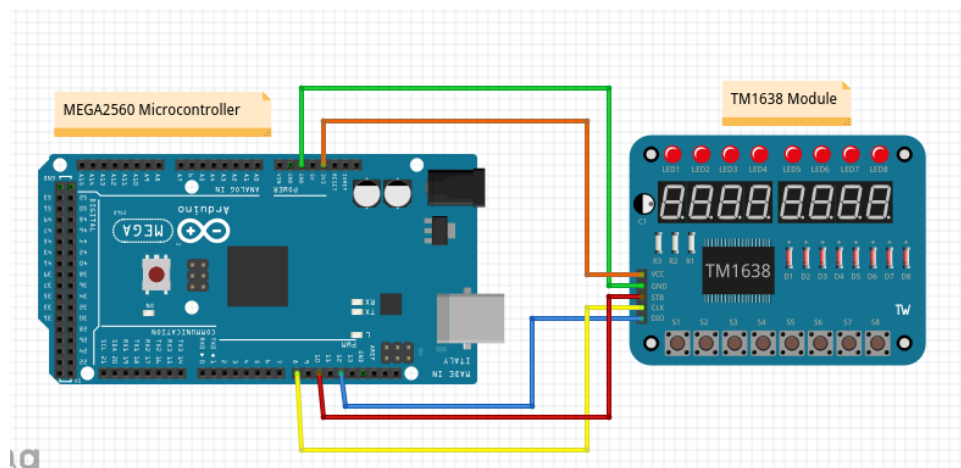


Figure 6 - Connect LED display

Whilst the data collected from the sensors has more use in powering graphs in place on the website, a simple display was implemented so that the user can see the real time data collected from the sensors. A TM1638 module was selected as it contains eight seven-segment displays that can be used to display basic letters and numbers and eight buttons which can be used to make selections. A TM1638 library was required to access the functionality of the buttons, via the module.getButtons() function, and allow data output to the seven-segment displays, via the module.setDisplayToString() function. The team decided to use the first four buttons to switch between which sensor reading was currently displayed, with the default display and button showing the temperature reading. The second button was set to display the humidity reading, third to display the light level and the fourth to display whether the plant required watering or not. First, all of the sensor values had to be converted to String objects as this is what the seven-segment displays required. A series of if statements was then used to interpret which button was activated and the appropriate reading would be displayed to the user

[see below code snippet]. The module also benefits from requiring only 3.3V to run, reducing the overall power draw of the system.

```

if(buttons==1)
{
  module.setDisplayToString("t "+tempString);
}
if(buttons==2)
{
  module.setDisplayToString("h "+humString);
}
if(buttons==4)
{
  module.setDisplayToString("l "+lightString);
}
if(buttons==8)
{
  if (m >= 600)
  {
    waterMe = "H2O PLS";
  }
  else if (m < 600)
  {
    waterMe = "NO H2O ";
  }
  module.setDisplayToString(waterMe);
}

```

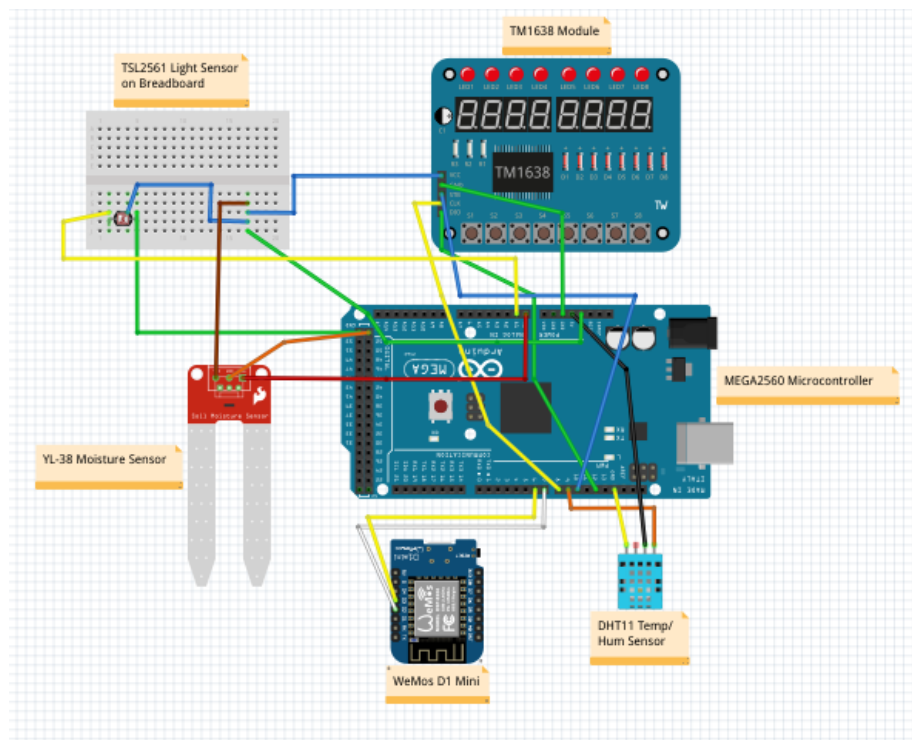


Figure 7 - Complete system circuit diagram

2.3 PROCEDURE PART 2 – WEBSITE

Designing an online platform, which would provide the means to monitor the plants status, began with establishing a list of functional and non-functional requirements that were expected to be fulfilled. First and foremost, the website aimed to present the collected readings in a graphical manner. That, in turn, may significantly facilitate optimization of plants health due to how important a role the environmental conditions play in gardening. Also, the platform is required to provide a way to register the device and the plant being monitored. By purchasing the product, the user

obtains access to an activation code assigned to the physical device. Logging into the system and adding the device is all it takes to begin utilising the system. Aside from the aforementioned features, the requirements regarding the Smart Eco Systems website included accessibility, as well as responsiveness, protection against malicious attacks and a visually attractive, user-friendly interface. Regarding the complexity of Smart Eco System, as well as the differences between the presentation and functional part of the website, it was decided that the most favourable solution would be splitting the platform into pre-login and post-login sections. By separating these logically independent parts, the company could make the most of the marketing opportunities created. Therefore, the front page, which is to some extent a business card of Abertay Plant Systems, was expected to provide basic information about the company and its innovative product. In order to organize the development process, all the requirements were converted into an extensive list of features, which were then allocated to sprints. Next, the first prototypes of both parts of the website were designed. Out of six front page versions designed by the sub team responsible for web development, the project manager decided to choose and implement three, which were considered as outstanding in terms of visual attractiveness and functional solutions. Although each of them made a good impression on paper, two of the implemented prototypes turned out to be potentially difficult to maintain and characterized by low responsiveness (Figure 8, Figure 9).

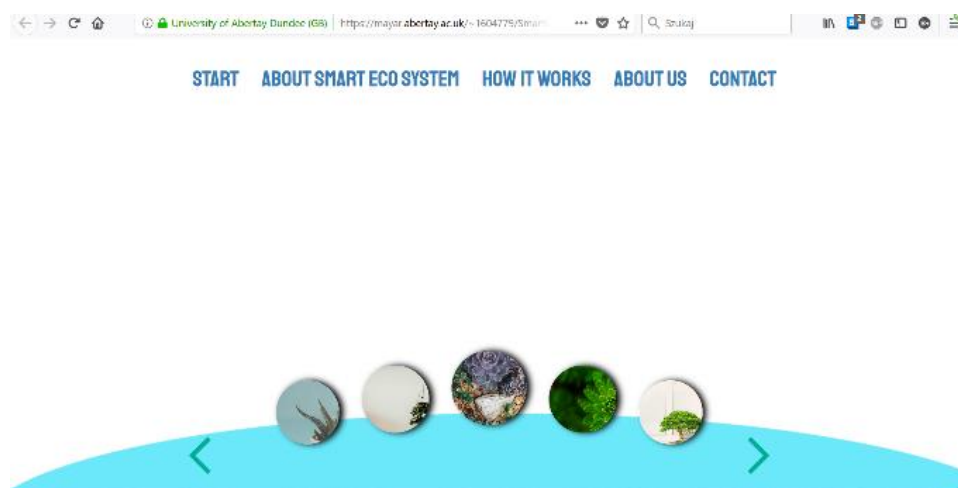


Figure 8 - Prototype rejected due to scalability problems of one of the features

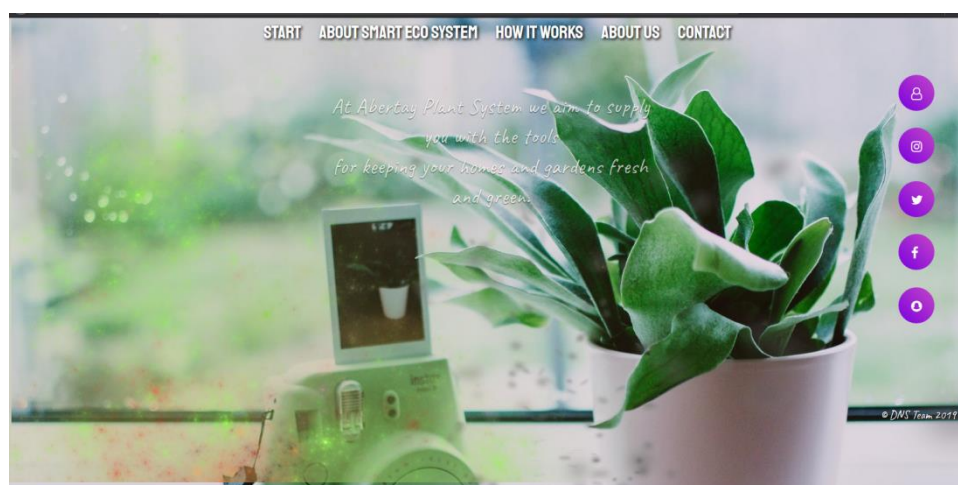


Figure 9 - Prototype rejected due to low level of responsiveness on mobile devices

Therefore, the prototype chosen combined the simplicity of expression with visually attractive, minimalistic layout adapted to different screen sizes. Having added a stylish font and high-quality

picture to the background, the prototype gained the desired look and could be extended with further elements.

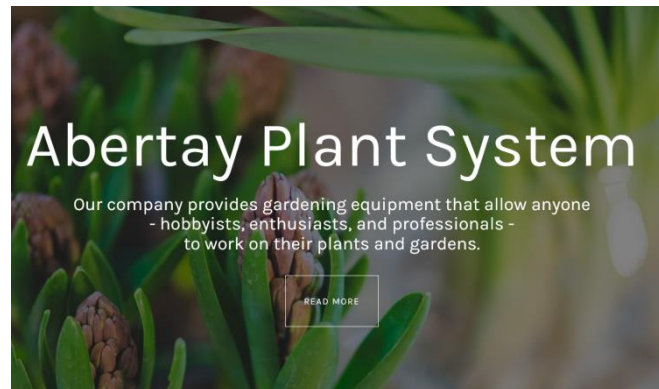


Figure 10 - Leading prototype of the front website

The development process began by enriching the application with the sliding mechanism using Bootstrap Carousel, which added dynamism to the minimalist design (Figure 10).

```
<script>
  var myIndex = 0;
  carousel();

  function carousel() {
    var i;
    var x = document.getElementsByClassName("mySlides");
    for (i = 0; i < x.length; i++) {
      x[i].style.display = "none";
    }
    myIndex++;
    if (myIndex > x.length) {myIndex = 1}
    x[myIndex-1].style.display = "block";
    setTimeout(carousel, 13000); // Change image every 4 seconds
  }
</script>
```

Figure 11 - Carousel mechanism (JavaScript)

The content of the landing page has been separated into five slides, which aimed at creating the effect of a digital brochure, directing the user to the most important section – contacting the producer (Figure 12).

```
<!-- Wrapper for slides -->
<div class="carousel-inner">

  <div class="item slides active">
    <div class="slide-1"></div>

    <div class="hero">
      <hgroup>
        <h1>Abertay Plant System</h1>
        <h3>Our company provides gardening equipment that allow anyone <br> - hobbyists, enthusiasts, and professionals -
        <br> to work on their plants and gardens. </h3>
      </hgroup>
      <button class="btn btn-hero" id="showText1">Read more</button>
    </div>
  </div>

  <div class="item slides">
    <div class="slide-2"></div>
    <div class="hero">
      <hgroup>
        <h1>Smart Eco System - Future of Gardening</h1>
        <h3>Our company provides gardening equipment that allow anyone <br> - hobbyists, enthusiasts, and professionals -
        <br> to work on their plants and gardens. </h3>
      </hgroup>
      <button class="btn btn-hero" id="showText2">Read more</button>
    </div>
  </div>
</div>
```

Figure 12 - Construction of the carousel

In order to allow the user to freely move between the slides, the team augmented the website with corresponding buttons, adapted both to desktop and mobile version (Figure 13). Next, the team decided to display the content allocated to each slide in an overlay in a way that would avoid fully covering of the main page, giving an impression of fluidity and lightness of the design.

```

<!-- Indicators -->
<ol class="carousel-indicators">
  <li data-target="#bs-carousel" data-slide-to="0" class="active"></li>
  <li data-target="#bs-carousel" data-slide-to="1"></li>
  <li data-target="#bs-carousel" data-slide-to="2"></li>
  <li data-target="#bs-carousel" data-slide-to="3"></li>
  <li data-target="#bs-carousel" data-slide-to="4"></li>
</ol>

```

Figure 13 - Slide indicators

Having implemented the additional layer, as well as filling in the white space with the proper content, a return option and a decorative element, the platform finally gained a professional character and following this, fulfilled its informative role (Figure 14).

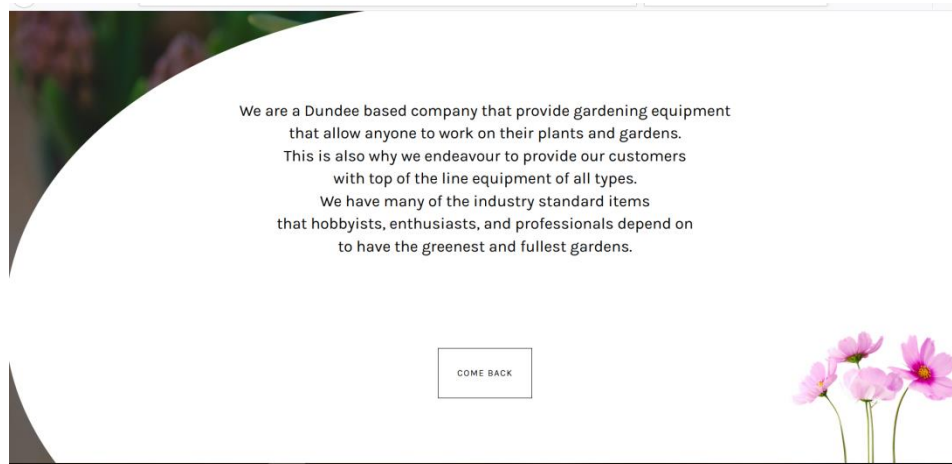


Figure 14 - Example content of the 'pop-up window'

Prior to moving to the implementation of the login & register facility, the team established a user table in a database, which would correspond to data required in the form. It consisted of a username and hashed password, the elements required for user authorization, as well as additional data such as email and full name.

	username	email	password	name	surname
▶	a	a@b	\$2y\$10\$WqLqQEztFtmMW/zVj/5kYOm.pHbtPR...	n	a
	aa	a@com	\$2y\$10\$gbxNgVG5ggMPb.QJQFDJK.uwH98ahr...		
	iza	iza@email.com	\$2y\$10\$cqIabH7nJNZWEE0Zixr0uUL25LRm8t...		
	j	j@j	\$2y\$10\$FI/G.WIN7FGgPaxVXwuJwuG522WmA4...	j	j
	Missmeva	m@email	\$2y\$10\$u/5iUJzherO7ViUdjQTieuhrU51rVY0RnI...		
	nader2929	n@email.com	\$2y\$10\$3sBgVU8PJss1sM9Xupb8YO4UD04GYz...	Nader	Sobhi
	scott	scott@email.com	\$2y\$10\$B8l6VJUDjnGp19pZokwVmeNZI8xBya0...		
	uyui	m@q	\$2y\$10\$a9oYGtD0scTJbz8zF1TtU.mDmuDSxOT...	iugfu	uiuy
✱	NULL	NULL	NULL	NULL	NULL

Figure 15 - User table (filled with test data)

Regarding the registration form, it was implemented by combining front end and back-end elements responsible for placing the data in the database. Once submitted, the data is sent with the POST method and is processed by the API with additional precautions, including parameterization of SQL queries (Figure 16).


```

$stmt = mysqli_stmt_init($conn);
$sql = "INSERT INTO plant_users(email, password, username) VALUES (?, ?, ?)";
mysqli_stmt_prepare($stmt, $sql);
mysqli_stmt_bind_param($stmt, 'sss', $email, $pwdhash, $username);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);
echo "<script>alert('Successfully registered');window.location.href = '../view/new2.php';</script>";

```

Figure 16 - Inserting the user into the database (API)

Though the main functionality of the landing page was implemented, one of the requirements indicated that the website be navigable using mobile touch rather than buttons. This feature was implemented using the swipe functions available from the jQuery library would be the simplest solution amid all available frameworks. (Figure 17).

```

1  $(function() {
2      $("#slides").swipe( {
3          swipe:function(event, direction, distance, duration, fingerCount, fingerData) {
4              if(direction=='right'){
5                  $('.carousel').carousel('prev');
6              }
7              if(direction=='left'){
8                  $('.carousel').carousel('next');
9              }
10         },
11         threshold:75
12     });
13 });
14

```

Figure 17 - JQuery function for swiping slides on touch (mobile devices)

The implementation of the members area of the website began with establishing a responsive, collapsible menu, which would allow the user to move to the desired section of the website with a single click on one of the available subsections. Moreover, the menu was designed to highlight the subsection currently in focus, as the usability of the platform was one of the main requirements (Figure 18).

```

var sections = $('section'), nav = $('nav'), nav_height = nav.outerHeight();

$(window).on('scroll', function () {
    var cur_pos = $(this).scrollTop();

    sections.each(function() {
        var top = $(this).offset().top - nav_height,
            bottom = top + $(this).outerHeight();

        if (cur_pos >= top && cur_pos <= bottom) {
            nav.find('a').removeClass('active');
            sections.removeClass('active');

            $(this).addClass('active');
            nav.find('a[href="#'+$(this).attr('id')+'"]').addClass('active');
        }
    });
});

```

Figure 18 - JQuery function for detecting the active element (subsection)

2.4 PROCEDURE PART 3 - DATABASE

Storing the data that is produced and used by the system is one of the main foundations that the application is built on. This is due to the fact that the application relies heavily on the ability to store

information for later use, the storage of data will mainly be performed by the Arduino System and the retrieval will be performed exclusively by the website. The format in which the data is stored is a combination of a Relational Database (RD) Model and a NoSQL model. In order to simulate the potential conditions that will be in place once the system goes into production it was decided to use a platform that could host the potentially considerable amount of data that will be collected. The platform that was deemed most appropriate was Amazon Web Services (AWS) which allows its users to lease all types of servers and server space ranging from simple RDs to servers setup to execute machine learning code. In our case a simple server setup to host a RD was all that was necessary. The setup process followed to lease the server space, create the database, and create the tables will be shown and described below.

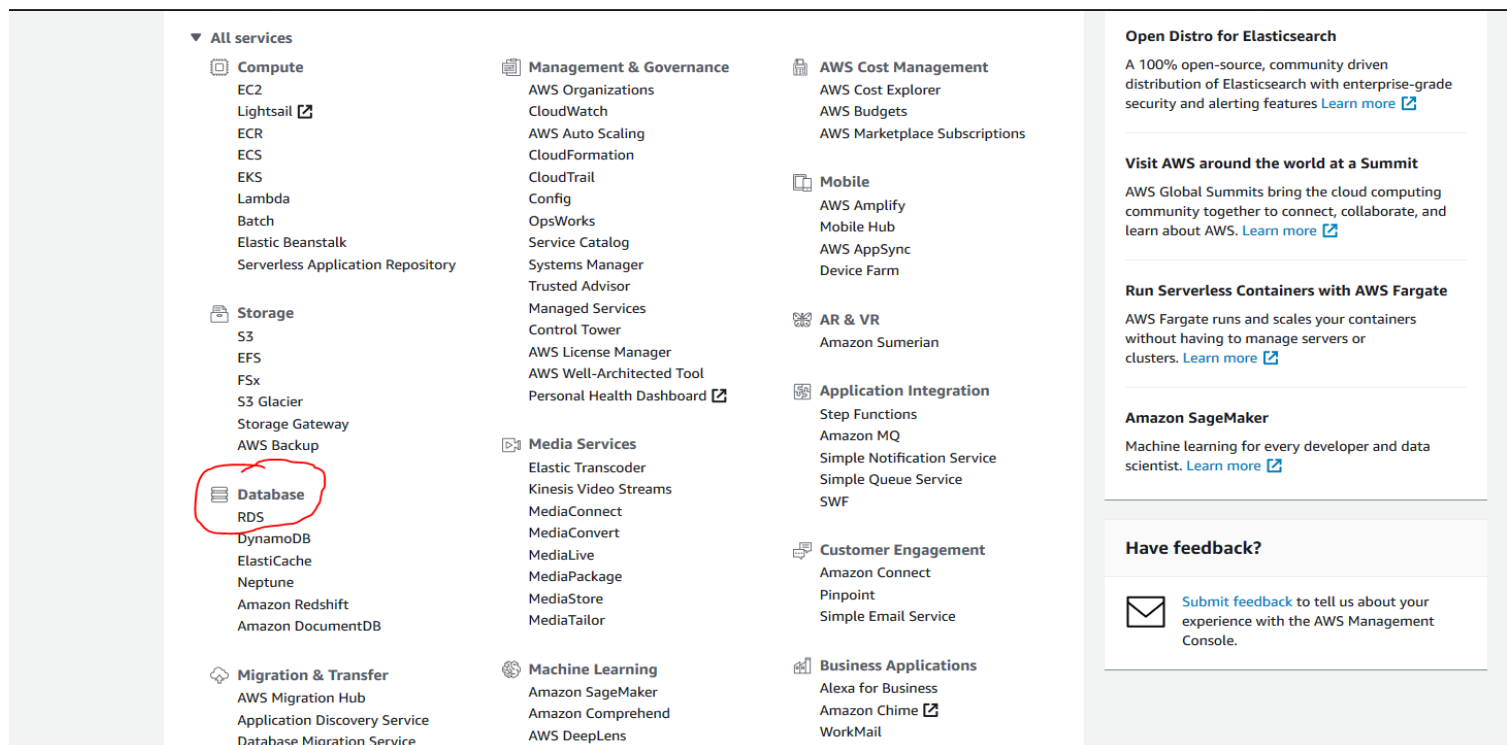


Figure 19

As seen in Figure 19 the user will be presented with this page once they sign up or sign in to AWS, from there they can select the type of service they want to use. In the case of this system an RDS instance was created. Following this the developer can then select the “Create Database” option, shown below in Figure 20, and follow the instructions displayed by the Setup Wizard.

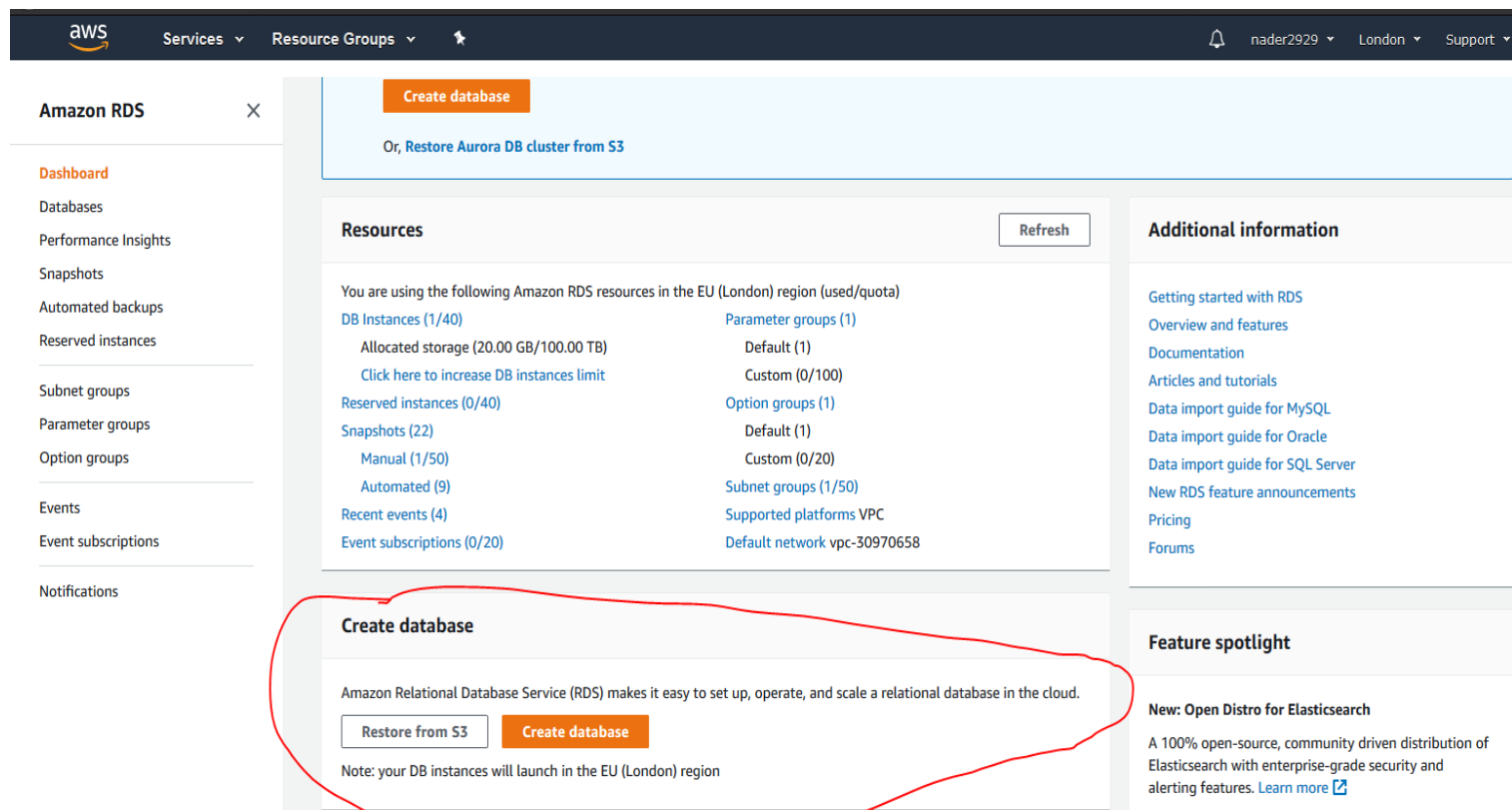


Figure 20

Once the setup was complete the user could select the instance they create and change any settings they wish using the panel provided, Figure 21 showcases this panel.

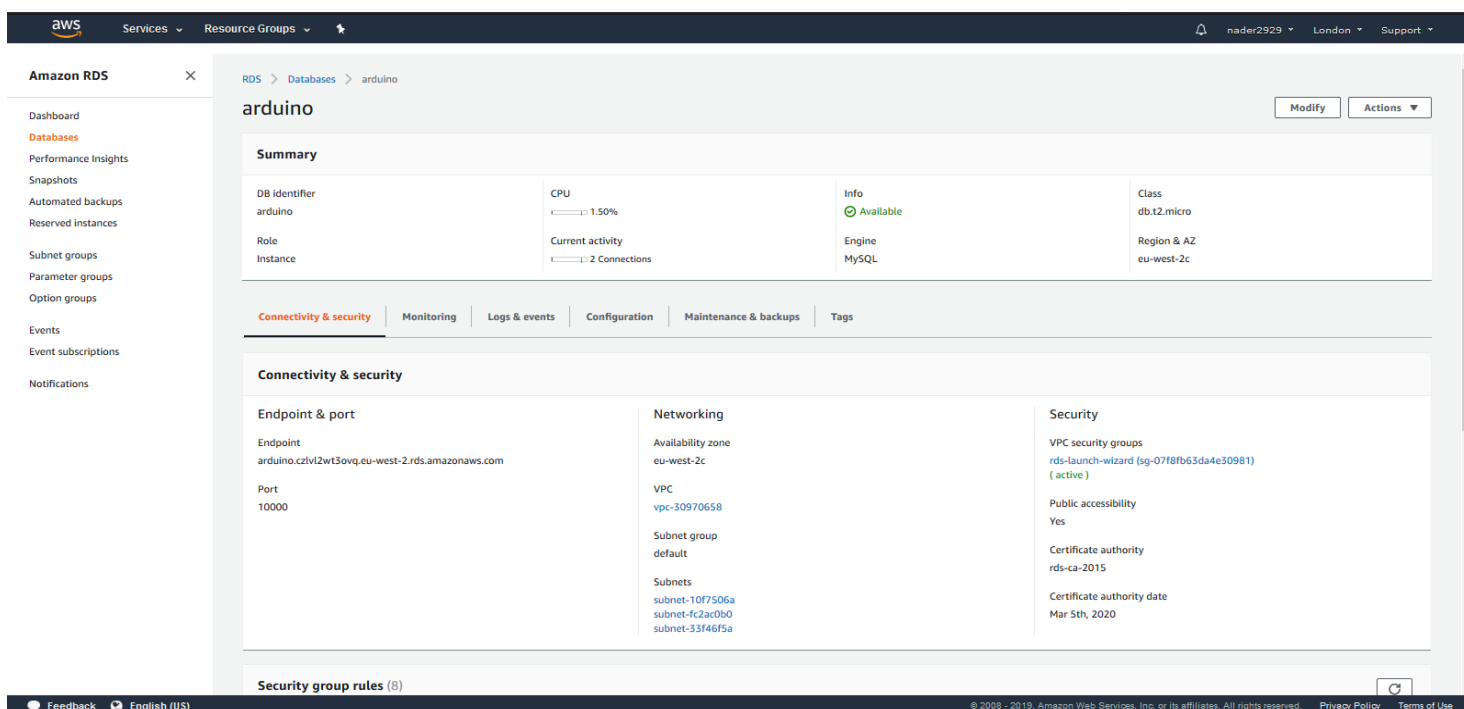


Figure 21

From there the user could use their choice of Database Management Software to create the tables required. In the case of this system the developer decided to use MySQL Workbench to implement the design. The first table that needed to be created was responsible for holding the readings recorded by the Arduino devices. The code used to achieve this is shown below in Figure 22.

```
CREATE TABLE `plant_readings` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `device` varchar(15) NOT NULL,  
  `time` timestamp(6) NOT NULL,  
  `readings` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=721 DEFAULT CHARSET=latin1
```

Figure 22

The table was then populated by the Arduino once it was operating correctly and could be used in the future by the website to present the readings in a graphical manner. The 'readings' column of this table is where the NoSQL aspect is evident. Due to the fact that the readings produced from the devices could in the future be extended, i.e. more sensors could be added to the device, this would not affect the database.

```
CREATE TABLE `plant_users` (  
  `username` varchar(35) NOT NULL,  
  `email` varchar(45) NOT NULL,  
  `password` varchar(600) NOT NULL,  
  `name` varchar(45) NOT NULL,  
  `surname` varchar(45) NOT NULL,  
  PRIMARY KEY (`username`),  
  UNIQUE KEY `email_UNIQUE` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Figure 23

In Figure 23 the code used to create the user table is shown. Once this table was created and the code that was responsible for populating the table was functional, users could easily login and register on the website.

```
CREATE TABLE `plants` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `device` varchar(100) NOT NULL,  
  `username` varchar(200) NOT NULL,  
  `image` varchar(300) NOT NULL,  
  `species` varchar(300) NOT NULL,  
  `requirements` varchar(300) NOT NULL,  
  PRIMARY KEY (`device`),  
  UNIQUE KEY `id` (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=latin1
```

Figure 24

In Figure 24 the code used to create the plants table, that would link the users to their device. It would then be populated by the users once they add their devices to their accounts.

3.1 RESULTS FOR PART 1

In this section, the results of the above outlined procedures will be discussed in three individual sections.

Results for Arduino

The manner in which the hardware system was put together allowed for results to be gained incrementally, i.e. once one peripheral had been attached, it was tested to ensure it was sending the correct data and then the next peripheral was attached and so on.

Arduino and WEMOS

The WEMOS unit was the first peripheral to be attached to the Arduino unit because ensuring a stable WiFi connection was imperative to the success of the project.

To test that the system worked, the Arduino was given a string that it passed to the WEMOS. If the WEMOS received the string, it would display it in the Serial Monitor of the Arduino IDE. Once successful, the destination of the string was then changed to the database to test the WiFi connection and whether the data was held correctly in the database.

This result was important as it gave the team a solid platform to build upon.

Arduino and Sensors

The sensors selected for the project were chosen not only for their ease of use regarding system implementation, but also on how they returned data to the system. For example, the DHT11 sensor was used to collect temperature and humidity readings. Not only did this combine two sensors in one but also returned readings in terms of degrees Celsius which meant that accurate results were being obtained as soon as the sensor was connected properly. This scenario was similar for the other sensors chosen in the project.

Testing of the sensors followed the same process as testing the WEMOS connection. The sensors would return values, and these would be displayed in the Serial Monitor. This allowed the team to fine tune how they wanted the data to be displayed as well as test that it was held correctly in the database.

Arduino and JSON String

The team decided that instead of storing the sensor readings in separate columns within the database, a JSON string would be used to collate the data and be held in one column of the database. The thinking behind this was based around how efficient the system could be at sending data as well as making it easier to parse this data for use on the website. Rather than sending four separate pieces of data to the database all at slightly different times, a JSON string is created that takes all four data 'chunks' and appends them sequentially. Once all data chunks have been received, the JSON string is then sent to and held in the database. This means the website only has to access three columns in the table (the ID of the hardware system, the time of entry and the JSON string) and can then parse out the required data from the JSON string where needed. This allows all four sensor readings to be sent at the same time, removing any discrepancies in the times of the collected data.

Arduino and LED Display

Whilst it was accepted that the main use of the data collected was to power the creation of graphs and charts on the website, the team decided that being able to view real-time data via a hardware display might also be useful. In order to achieve this, the TM1638 module was selected because it features a display and switch buttons on the same unit.

The display itself is made up of eight 7-segment LEDs, a rather simple form of display that was perfect for displaying numbers, which would mainly be the datatype on show. However, the display can have problems displaying certain letters of the alphabet, 'M' and 'W' being prime examples. Four readings had to be displayed on the screen, namely temperature, humidity, light level and moisture level. To ensure that the user knows which sensor data is being displayed, the first LED displays a lowercase letter relating to the sensor type; 't' for temperature, 'h' for humidity and 'l' for light. Displaying the moisture level was more problematic, in that while the moisture level could be displayed, it could not be labelled similar to the previous displays. To overcome this, the code was modified slightly so that the display reads 'H2O PLS' if the plant requires watering or 'NO H2O' if watering isn't required. Whilst the display isn't uniform in how it communicates the real time data to the user, the display device has been kept as the TM1638 module as it is simple to use, displays data accurately and draws very low power compared to more complex display modules.

Results for Website

The website is the main interface that the user will have to interact with in order to fully utilise the system, so it was necessary to thoroughly test the website to ensure that it was functional, easy to use and secure.

The facets of the system that were tested are going to cover multiple aspects, mainly the functionality of the website and mobile testing. More info on each aspect, as well as their results, will be given below. The overall website will also be evaluated for its browser compatibility as well as its speed.

The tests for this aspect of the system were performed by anonymous users as well as by the developers, and the feedback produced was used to make modifications where appropriate.

Login and Register

Functionality Testing

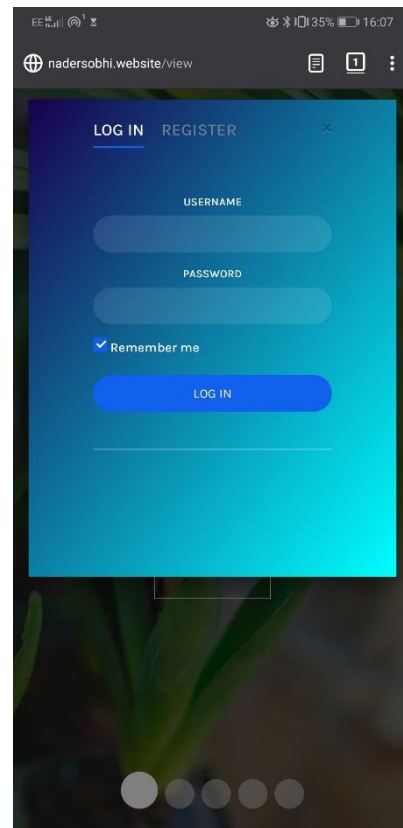
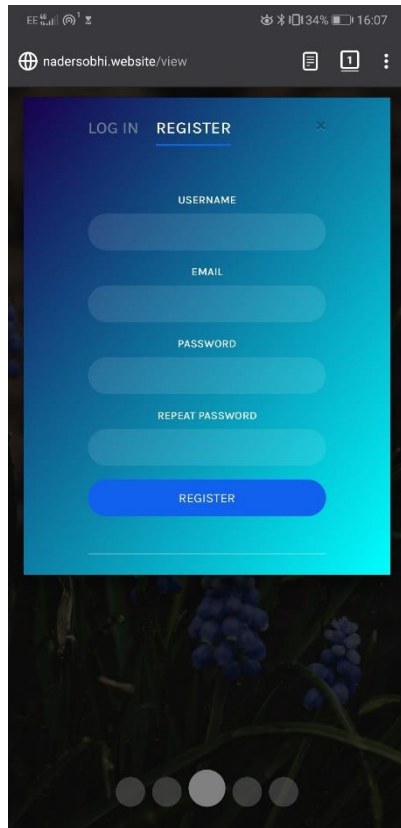
This is the testing aspect that will affect users the most as it is a gauge of whether or not users can perform the actions they need using the platform created.

After experimenting with the website, the testers indicated that while the website works as it should, there is no need for storing the name and surname of the person registering. Upon receiving this feedback, the developers removed those input fields from the form used for registering. This also meant that the application would be handling a smaller amount of sensitive data and therefore worries about security breaches are reduced.

Mobile Testing

With around 50% of the world's internet traffic being served to mobile devices it was vital to ensure that users could easily use the web application from mobile devices.

Once the login and register functions were tested on mobile it was deemed that they implemented in such a way that was suitable for mobile devices, as shown in the screenshots below.



Front Page Navigation

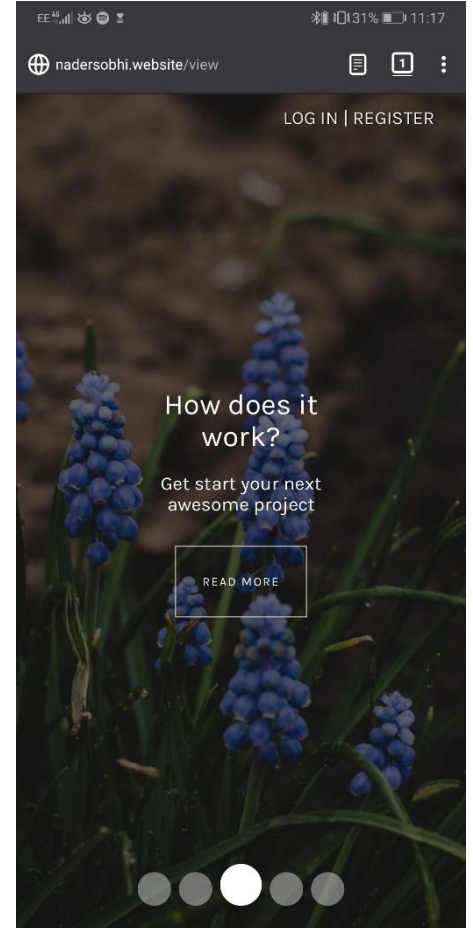
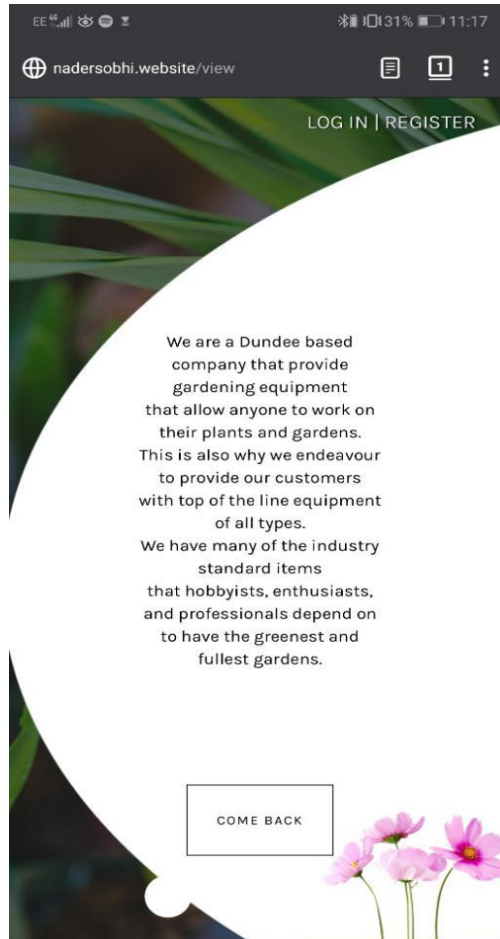
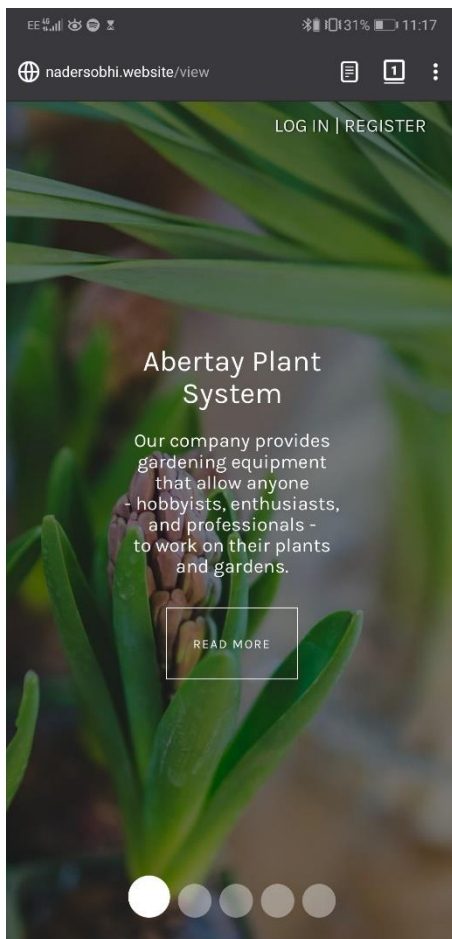
The page that greets the user upon accessing the website had to be functional as well as attractive to use.

Functionality Testing

After allowing the testers to use the website's landing page and try to casually browse the information presented in order to get a feel for the application, some feedback was generated. Testers stated that they found the site easy to use and browse. Testers also noted that they really enjoyed the smoothness with which the front page allows the users to navigate between the different slides.

Mobile Testing

To ascertain whether or not users could easily navigate the web app on mobile devices they were asked to try and perform the same action they had done on the desktop version of the website on a mobile device. Once they had done so the developers received feedback, it was determined that the testers found the front page a delight to navigate as it felt like an app more than a website. This was encouraging to the developers as that was the goal they set out to achieve when creating the web app. Example images of what the website looks like on mobile are shown below.



Plants Panel Navigation

Once the user has logged in they will be redirected to a panel that will allow them to view the readings that their device(s) has collected. They will also be able to add more devices should they need to.





Functionality Testing

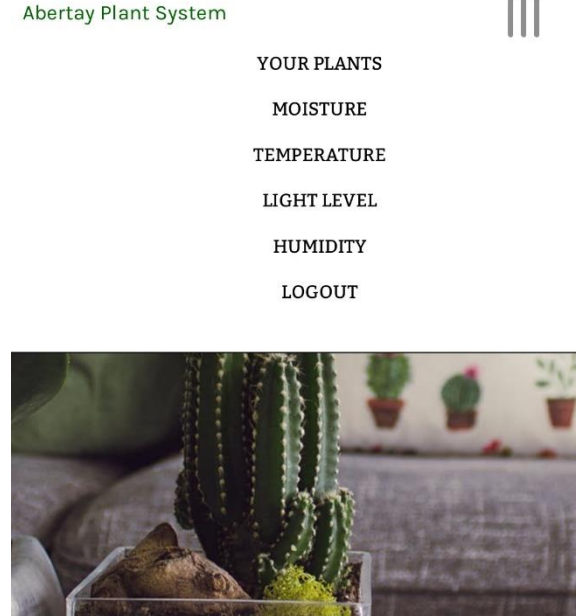
The main feature that was tested was the navigation between the different sections and graphs, as well as the functionality of the following features: Adding devices and plants to the users' accounts, Switching the graphs view from a Day view to a Month or Week view, and using the back to top button. Once the testers tried the features and casually browsed the page to get a feel for the ease of navigation of the webpage, they gave their feedback to the developers. The overall feedback was that the website was easy to browse, and the features were intuitive to use.





Mobile Testing

Making this panel functional and usable on mobile was vital as it would allow users to check the status of their plants using their mobile device. The feedback received was that while the website is functional and enjoyable to use on mobile devices, the font size on the graphs is rather small. This was noted down but not changed due to the fact that changing this might negatively affect the desktop version of the website. Examples of what the Plant Panel page looks like on mobile are shown below.



	Orchid 5C:CF:7F:10:2D:	
	Roses 5C:CF:7F:10:4D:	



	Orchid	5C:CF:7F:10:2D:	 Too much water
	Roses	5C:CF:7F:10:4D:	

Overall website tests

The website was then tested for its overall browser compatibility and speed. The reason these attributes were separated from the features tested above was that these attributes contribute to the overall feel of the website and not specific to any page or feature.

Browser Compatibility

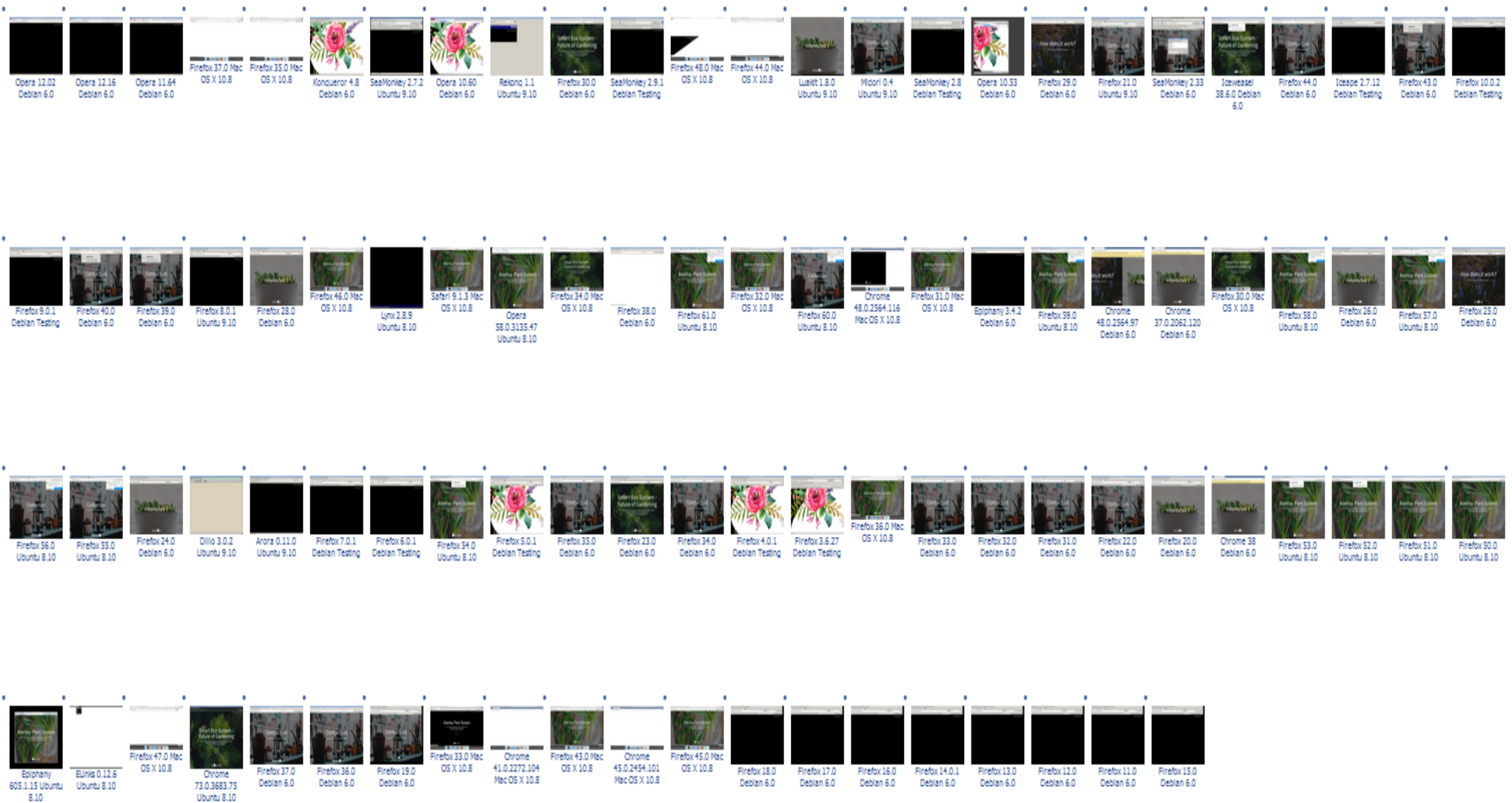
In order to test the website on multiple browsers in a timely and efficient manner the developers used an online service provided by browsershots.org. This service allows the developer to view screenshots of any website they choose on 95 different browsers. These screenshots are shown in the next page. As seen below, all modern browsers and a lot of the older versions of popular browsers seem to function well.

Screenshot request group 1

[Hide Status](#) [Download All](#)

Submitted 53 minutes ago

95 browsers selected, 95 uploaded



Speed

Another measure of responsiveness is the speed with which the website loads and responds to the users clicks and touches. There are a number of facilities online that allow a developer to view their websites performance, for the purposes of this website Google PageSpeed Insights and KeyCDN Speed Test were used. The results of these test are shown below.



http://nadersobhi.website/

The **speed score** is based on the lab data analyzed by [Lighthouse](#).

Analysis time: 19/04/2019, 15:18:12

Scale: ● 90-100 (fast) ● 50-89 (average) ● 0-49 (slow)

Field Data

The Chrome User Experience Report [does not have sufficient real-world speed data](#) for this page.

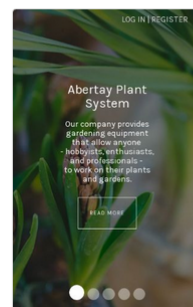
Origin Summary

The Chrome User Experience Report [does not have sufficient real-world speed data](#) for this origin.

Lab Data

[Lighthouse](#) analysis of the current page on an emulated mobile network. Values are estimated and may vary.

First Contentful Paint	3.4 s ⓘ	First Meaningful Paint	3.5 s ⓘ
Speed Index	9.0 s ▲	First CPU Idle	4.7 s ⓘ
Time to Interactive	6.3 s ⓘ	Estimated Input Latency	10 ms ✓



URL	Status	Size	Time	Timeline
http://nadersobhi.website/	200	53 B	9.6 s	
view http://nadersobhi.website	301	0 B	151.563 ms	
http://nadersobhi.website/view/	200	22 KB	159.681 ms	
bootstrap.min.css https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css...	200	118 KB	3.1 s	
newcss.css http://nadersobhi.website/view	200	9 KB	158.685 ms	
jquery.min.js https://ajax.googleapis.com/ajax/libs/jquery/3.3.1	200	85 KB	3.1 s	
jquery.mobile.js https://cdnjs.cloudflare.com/ajax/libs/jquery-mobi...	200	455 KB	3.2 s	
bootstrap.min.js https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js	200	36 KB	3.1 s	
font-awesome.min.css https://cdnjs.cloudflare.com/ajax/libs/font-awesom...	200	30 KB	3.2 s	
css?family=Titillium+ Web:400,300,600 https://fonts.googleapis.com	200	2 KB	3.1 s	
jquery.touchSwipe.min.js https://cdnjs.cloudflare.com/ajax/libs/jquery.touc...	200	20 KB	3.2 s	
?u=http%3A%2F%2Fwww.pngpix.com%2Fwp-content... cont... https://proxy.duckduckgo.com/iu	200	1 MB	302.487 ms	
?u=http%3A%2F%2F3.bp.blogspot.com%2F6Y0o... cont... https://proxy.duckduckgo.com/iu	200	198 KB	308.328 ms	
arduino.png http://nadersobhi.website/view	200	554 KB	1.2 s	
sensor.png http://nadersobhi.website/view	200	688 KB	1.4 s	

Upon analysis of the results it is clear that the website is not performing optimally in terms of speed. This is mainly due to the images used on the website which use old, outdated formats, as shown below.

 Opportunities

These optimizations can speed up your page load.

Opportunity		Estimated Savings	
1	Serve images in next-gen formats	<div><div></div></div>	24.75 s
2	Preload key requests	<div><div></div></div>	4.8 s
3	Eliminate render-blocking resources	<div><div></div></div>	1.58 s
4	Minify JavaScript	<div><div></div></div>	0.3 s
5	Properly size images	<div><div></div></div>	0.15 s
6	Remove unused CSS	<div><div></div></div>	0.15 s

Unfortunately, these images are vital to the visuals of the website and therefore cannot be changed, a potential future change would be to convert these images to newer, more efficient formats.

Results for database

The database used to hold the data acts as a type of bridge between the website and the device so naturally the developers needed to ensure that it was working correctly and securely. Luckily, the testing and the subsequent results for the database are minimal as all that needs to be ensured is that the database cannot be accessed from the backend. In this section the backend security of database will be discussed and shown.

The actions that the developers took was to set a unique username and password for the database that differed vastly from the default options, as well as switching the port that the database operates under from the default 3306 to port 10000. These changes are shown below in Figure 25 with the password being redacted by the setup screen of MySQL Workbench.

The screenshot shows the 'Setup New Connection' dialog in MySQL Workbench. The 'Parameters' tab is active. The 'Connection Name' field is empty with the hint 'Type a name for the connection'. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' section contains: 'Hostname' (2wt3ovq.eu-west-2.rds.amazonaws.com), 'Port' (10000), 'Username' (root), 'Password' (redacted with 'Store in Vault ...' and 'Clear' buttons), and 'Default Schema' (empty). The 'Test Connection' button is highlighted in blue.

Figure 25

In order to showcase the databases ability to hold data in an organized manner screenshots of some of the data held in each table will be shown below.

	id	device	time	readings
▶	2	5C:CF:7F:10:4D:	2019-03-05 18:14:06.000000	{"m": "689", "h": "25.00", "t": "2", "l": "678"}
	3	5C:CF:7F:10:4D:	2019-03-05 19:07:52.000000	{"m": "689", "h": "2", "t": "29.00", "l": "101"}
	4	5C:CF:7F:10:4D:	2019-03-05 19:07:58.000000	{"m": "689", "h": "26.00", "t": "29.00", "l": "102"}
	5	5C:CF:7F:10:4D:	2019-03-05 19:08:08.000000	{"m": "689", "h": "25.00", "t": "28.00", "l": "108"}
	6	5C:CF:7F:10:4D:	2019-03-05 19:08:17.000000	{"m": "689", "h": "27.00", "t": "28.00", "l": "100"}
	7	5C:CF:7F:10:4D:	2019-03-05 19:08:26.000000	{"m": "690", "h": "28.00", "t": "28.00", "l": "104"}
	8	5C:CF:7F:10:4D:	2019-03-05 19:08:35.000000	{"m": "689", "h": "28.00", "t": "28.00", "l": "109"}
	9	5C:CF:7F:10:4D:	2019-03-05 19:08:44.000000	{"m": "689", "h": "28.00", "t": "28.00", "l": "106"}
	10	5C:CF:7F:10:4D:	2019-03-05 19:08:53.000000	{"m": "689", "h": "27.00", "t": "28.00", "l": "105"}
	11	5C:CF:7F:10:4D:	2019-03-05 19:09:02.000000	{"m": "689", "h": "27.00", "t": "28.00", "l": "113"}
	12	5C:CF:7F:10:4D:	2019-03-05 19:09:12.000000	{"m": "689", "h": "28.00", "t": "28.00", "l": "105"}
	13	5C:CF:7F:10:4D:	2019-03-05 19:09:21.000000	{"m": "690", "h": "27.00", "t": "28.00", "l": "677"}
	14	5C:CF:7F:10:4D:	2019-03-05 19:09:30.000000	{"m": "690", "h": "24.00", "t": "28.00", "l": "678"}
	15	5C:CF:7F:10:4D:	2019-03-05 19:09:39.000000	{"m": "690", "h": "23.00", "t": "28.00", "l": "677"}
	16	5C:CF:7F:10:4D:	2019-03-05 19:09:48.000000	{"m": "690", "h": "22.00", "t": "28.00", "l": "678"}
	17	5C:CF:7F:10:4D:	2019-03-05 19:09:57.000000	{"m": "690", "h": "23.00", "t": "28.00", "l": "679"}
	18	5C:CF:7F:10:4D:	2019-03-05 19:10:07.000000	{"m": "690", "h": "23.00", "t": "28.00", "l": "679"}

Figure 26

Figure 26 - shows the *plant_readings* table columns as well as some of the records held in the table.

	username	email	password	name	surname
▶	a	a@b	\$2y\$10\$WqLqQEztFtmMW/zVj/5kYOm.pHbtPR...	n	a
	aa	a@com	\$2y\$10\$gbxNgVG5ggMPb.QJQFDJK.uwH98ahr...		
	iza	iza@email.com	\$2y\$10\$0cqIabH7nJNZWEE0ZiXr0uUL25LRm8t...		
	j	j@j	\$2y\$10\$FI/G.WlN7FGgPaxVXwuJwuG522WmA4...	j	j
	Missmeva	m@email	\$2y\$10\$u/5iUJzherO7ViUdjQTieuhrU51rVY0RnI...		
	nader2929	n@email.com	\$2y\$10\$3sBgVU8PJSs1sM9Xupb8YO4UD04GYz...	Nader	Sobhi
	scott	scott@email.com	\$2y\$10\$88l6VJUDjnGp19pZokwVmeNZI8xBya0...		
	uyui	m@q	\$2y\$10\$a9oYGT0scTJbz8zF1TtU.mDmuDSxOT...	iugfu	uiuy
*	NULL	NULL	NULL	NULL	NULL

Figure 27

Figure 27 - shows the *plant_users* table columns as well as some of the records held in the table.

	id	device	username	image	species	requirements
	11	5C:CF:7F:10:2D:	iza	../view/images/proxy.duckduckgo.com.jpg	Orchid	{"light": "1", "water": "1", "temperature": "1"}
▶	6	5C:CF:7F:10:3D:	Missmeva	../view/images/Internet.jpg	Daffodil	{"light": "1", "water": "1", "temperature": "1"}
	10	5C:CF:7F:10:4D:	iza	../view/images/testmount.jpg	Roses	{"light": "1", "water": "2", "temperature": "3"}
*	NULL	NULL	NULL	NULL	NULL	NULL

Figure 28

Figure 28 - shows the *plants* table columns as well as some of the records held in the table.

4.1 GENERAL DISCUSSION

This section will look at the project overall, how it relates to similar solutions already on the market and how it meets the aims set out in a previous section.

The project has been very successful in terms of producing a hardware and software system that accurately collects, records and displays data about a specific plant. Whilst on paper this may seem like a trivial exercise, translating this into a system that is robust enough to deal with changing variables while collecting useful information is quite another challenge. Care had to be taken to ensure that the data collected from the sensors was in a human readable and useful format before it was sent to the database. The database itself had to be designed in such a way that the website could access it easily and simply extract the data it required. The website then had to use this data to power graphs and charts that the user could use to track the status of their plant over a given time frame.

It is important to evaluate the project in regards to similar devices currently on the market. The Plant Monitoring System created by Ryan Gill uses similar components to achieve a similar result, but it relies on slightly inflexible open source software to collect and render the data. The system developed by the team has enough scope that sensors can be added with only slight alterations to the code base. Gill's system also requires technical expertise to put together, a factor the developed system negates. The total cost of Gill's system is slightly less than the team's system, but any potential saving is destroyed by the technical knowledge required to build it.

The Smart Plant Watering System by Elecrow again uses similar components, but adds in the feature of remote plant watering. However, this not only greatly inflates the cost to be over £100 more expensive, but the technical knowledge required to build it is vast. This system also avoids any data collection and uses only moisture sensors to coordinate its operations. However, the inclusion of remote plant watering makes it a very interesting device and should be considered in future iterations of the developed device.

The Smart Garden System by SwitchDoc is perhaps the industry leader in regards of smart plant monitoring. It uses a Raspberry Pi in place of an Arduino, which gives it much more processing power and the ability to add sensors and modify the code. However, it requires some technical knowledge to put together, takes up a much larger footprint and is also much more expensive than the developed device. The ability to monitor more than one plant from the same 'brain' is a highlight, as is its integration with digital assistants like Amazon's Alexa.

All of the devices mentioned do very similar things as the developed device as well as features beyond the scope of this project. However, they all suffer from the same flaw: they require specialized knowledge to build, operate and maintain. This is where the developed device stands on its own.

In a previous section, the aims of the project were laid out. It is important to evaluate how well the project meets those aims discussed. The high-level aim that was set out stated that a product should be created that met the needs of the modern gardener, both those just discovering the field of horticulture as well as more seasoned individuals. The goal here was to develop a system that was not only easy to setup and use, but also one that would feedback useful information to the user. Overall, the project achieves this aim well. The physical device has been developed to include as little peripheral modules as possible whilst still maintaining a high degree of usability. To set up the device, the user has only to place the sensors where they feel is most accurate and then power the device on. Inserting data into the database is done automatically upon a WiFi connection being established. The user then has to register with the companion website to view the graphs and charts powered by the collected data. The website has been designed for use with mobile and desktop devices in mind.

In addition to this over-arching aim, several smaller aims were identified. Firstly, the website had to be responsive in order to display correctly on a mobile device as well as desktops, laptops and tablets.

Consideration on which browsers being used played a part here, as older users may be prone to continued use of legacy browsers, such as Internet Explorer. Navigation and content of the website was also a crucial consideration, with the aim of ensuring as few button clicks as possible resulted in the user locating the desired information. Secondly, the device had to be easy to maintain with little or no expert knowledge required, as well as perform in a reliable manner. The developed hardware device is essentially 'plug and play', with the user having merely to place the sensors and power the device on. The components used are rugged enough to withstand repeated use but also simple in their operation. The reliability of the device is dependent on the WiFi connection. If a user's WiFi service was to be interrupted, sensor readings may not be placed in the database. This, however, is out with the scope of the team but in attempt to lessen such an impact, sensor data is only sent to the database every thirty minutes. Lastly, collection of data had to be undertaken in a manner appropriate with current legislation, with consideration being given to how these laws may change in the future. While the hardware device is geared toward collecting data, the only piece of personal data the user has to provide is their email address. This is so that they can register a username and password for use on the site with the email address only being used should they request to reset their password. However, the website does contain a 'Remember Me' function that uses cookies to store session information. To inform the user of this, a pop-up is generated when they visit the site to inform them that cookies are used. After agreeing with this, they are free to use the website. This follows GDPR legislation dealing with the collection of cookies.

It is fair to say that the overall project meets the needs of the client as well as the aims set out. The client wanted an intuitive, adaptable system that could collect, record and store data on a given plant. This was achieved. The aims then expanded on the client's needs to ensure the device was reliable, responsive and easy to use. This was also achieved.

4.2 CONCLUSIONS

Smart Eco System is a unique product, designed to help the client attract a large cross section of society. Having access to the readings presented in the form of visually attractive graphs, as well as the indicators showing a plant's demand for water, all they need to do is to regularly check its status by logging into the online platform or using the attached display.

Moreover, such a solution could potentially raise the prestige of the company and significantly contribute towards the development of the business, and thus increase the number of connections in the industry. It would change the perception of the company from a small, local shop to competition others have to reckon with. That, in turn, would translate into bigger profits due to their new-found reputation as a pioneer of smart home equipment.

However, if the client, for some reason, decided not to release the product, the aforementioned competitors would definitely take advantage of the company's stagnation and introduce interesting technological solutions.

4.3 FUTURE WORK

As with any project created to fit a client's specification there are some features or aspects of the system that were identified as potentially beneficial but were not implemented due to some extenuating circumstances.

Starting with the database, a NoSQL solution could have been implemented instead of the current model which consists of a mix of Relational Database Scheme (RDS) and a NoSQL scheme. The reasoning for this was that the information stored currently would only consists of some basic user information, the sensor

readings from the device(s) and nothing else. However, in the future it may be prudent to switch the storage method to a NoSQL solution as it would allow for the easy expansion of any of the data-centric features.

The physical device responsible for collecting readings on a plants surroundings could be improved in a multitude of ways. These are mainly quality of life changes and therefore were not implemented at this point in time. Beginning with the implementation of some sort of interface for the Arduino Subsystem so as to allow the user to perform a few actions that are currently unavailable or only available through the web application. As far as the developer can identify there are only 2 ways that an interface could be implemented to allow the user to control the device's functionality. The first would be a custom-built application that would interface with the device once it is connected to the user's personal computer. This version would allow the user to make changes to the device's functions through the interface and then the appropriate instructions would be loaded onto the device. This iteration would be the simplest to implement as the Windows OS can already recognize when an Arduino device is connected. The other method that could be used to implement an interface would be to use the website to allow the user to set some options that would control the functionality of the device and then load the appropriate instructions to it. This is probably the hardest interface to implement as it requires that two-way communication is possible between the device and the website. Additionally, it would require more storage for user preferences as well as more data to be collected on each individual user. Lastly and least importantly at this stage would be the design and construction of a plastic 3D printed case for the device to be housed. This was not carried out due to the extremely limited access the developer had to such a printer.

The website is the final component which holds potential for future upgrades or modifications. Another security feature that will be implemented in the future will be the use of the HTTPS protocol to encrypt connections between the users and the website. This was not implemented because the web hosting used did not support the developers preferred method of adding the required certificate to the website and thus would require more work and time than available. Apart from security features there are some functional aspects that have not been implemented due to time constraints as well as the fact that they were deemed as small quality of life changes and therefore would not improve the functionality of the system by a significant amount. That being said, they will be discussed here as potential improvements for the system in the future. Starting with the implementation of a feature that would enable users to discuss plant care with plant experts or members of the user base who have proven to be extremely adept at plant care. This would allow for a kind of social network that would enable users to interact with other people who share the same passions as them. Finally, another feature that could be implemented in the future would be to add a "wiki" type section of the website that would host all kinds of information about plant care. While very easy to implement this can be very time consuming as it would require the developer in charge of the task to do extensive research so as to ensure that the information hosted on the website is accurate.

4.4 CALL TO ACTION

To find out more about how you can gain the power to track and optimize your plants lifecycle with the Abertay Plant Monitoring System, visit www.abertaypms.com today to arrange a free, in-depth demonstration. The first fifty people to sign up for a demonstration session will receive 25% off the total cost of purchasing the system and six months of ongoing training and technical support. The first hundred people to sign up will receive six months of ongoing training and technical support. The only way to take advantage of these offers is to visit www.abertaypms.com today and register your interest in the Abertay Plant Monitoring System!

References

- Article title:** What the 'Internet of Things' Means For Small Business
Website title: Smallbusinesscomputing.com
URL: <https://www.smallbusinesscomputing.com/News/Networking/what-the-internet-of-things-means-for-small-business.html>
- Article title:** Accessibility - W3C
Website title: W3.org
URL: <https://www.w3.org/standards/webdesign/accessibility>
- Article title:** Project Planning a Step by Step Guide
Website title: Project Smart
URL: <https://www.projectsmart.co.uk/project-planning-step-by-step.php>

APPENDIX A - CODE

WEMOS Code

```
#include <ESP8266WiFi.h>

#include <ESP8266mDNS.h>

#include <WiFiUdp.h>

#include <WiFiUdp.h>

#include <WiFiClient.h>

#include <SoftwareSerial.h> //Include the NewSoftSerial library to send serial commands to the cellular module.

#include <string.h> //Used for string manipulations

#include <ArduinoJson.h>

#include <MySQL_Connection.h>

#include <MySQL_Cursor.h>

const char* ssid = "AndroidAP";

const char* password = "nader123";

WiFiClient client;

MySQL_Connection conn((Client *)&client);


IPAddress server_addr(35,178,39,124); // MySQL server IP,

char user[] = "nader"; // MySQL user
```

```

char sqlpassword[] = "arduinoproject";

WiFiServer server(80);

char incoming_char = 0; //Will hold the incoming character from the Serial Port.
//StaticJsonDocument<1024> jsonBuffer;
DynamicJsonDocument jsonBuffer(2048);
SoftwareSerial cell(D2,D3); //Create a 'fake' serial port. Pin 2 is the Rx pin, pin 3 is the Tx pin.
bool moisture, humidity, light, temp = false;
char m='X'; char h='X'; char l='X'; char t='X';
int mvalue1=0; int hvalue1=0; int lvalue1=0; int tvalue1=0;
String mresult="";String lresult="";String tresult="";String hresult="";
int count = 0;
char jsonstr;
void setup()
{
  //Initialize serial ports for communication.
  Serial.begin(9600);
  cell.begin(9600);
  Serial.print(WiFi.macAddress());
  //Let's get started!
  Serial.println(" Starting SM5100B Communication...");
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("No connect ");
  }
  server.begin();
  while (conn.connect(server_addr, 10000, user, sqlpassword) != true) {
    Serial.print ( " sql connect failed " );
  }
}

void loop() {
  char check;

```

```

char value;

//If a character comes in from the cellular module...
if(cell.available())
{
    incoming_char=cell.read();
    if (incoming_char == 'L'){
        if (lresult!=""){
            Serial.println("L"+lresult);
        }
        lresult="";
        Serial.println();
        count=count+1;
        l='L';
        check = incoming_char;
        moisture=false;
        humidity=false; light=true; temp = false;
    }
    else if (incoming_char == 'M'){
        if (mresult!=""){
            Serial.println("M"+mresult);
        }
        mresult="";
        Serial.println();
        count=count+1;
        m='M';
        check = incoming_char;
        humidity=false;
        moisture=true; light=false; temp = false;
    }
    else if (incoming_char == 'H'){
        if (hresult!=""){
            Serial.println("H"+hresult);
        }
        hresult="";

```

```

Serial.println();

count=count+1;

h='H';

check = incoming_char;

temp=false;

moisture=false; light=false; humidity = true;
}

else if (incoming_char =='T'){

    if (tresult!=""){

        Serial.println("T"+tresult);

    }

    tresult="";

    Serial.println();

    count=count+1;

    t='T';

    check = incoming_char;

    light=false;

    moisture=false; humidity=false; temp = true;

}

else{

    value = value+incoming_char;

    if (moisture==true){

        mresult=mresult+value;

        mvalue1=mvalue1+1;

    }

    else if (humidity==true){

        hresult=hresult+value;

        hvalue1=hvalue1+1;

    }

    else if (light==true){

        lresult=lresult+value;

        lvalue1=lvalue1+1;

    }

}

```

```

else if (temp==true){
    tresult=tresult+value;
    tvalue1=tvalue1+1;
}
}

if (mvalue1>=6 && tvalue1>=6 && lvalue1>=6 && hvalue1>=6){

    if (mresult=="" && hresult=="" && tresult=="" && lresult==""){
        digitalWrite(11, LOW);
    }

    jsonBuffer["m"]=mresult;
    jsonBuffer["h"]=hresult;
    jsonBuffer["t"]=tresult;
    jsonBuffer["l"]=lresult;

    serializeJson(jsonBuffer, Serial);

    String jsonstr="";

    serializeJson(jsonBuffer, jsonstr);

    MySQL_Cursor *cur_mem = new MySQL_Cursor(&conn);

    String q = "INSERT INTO DDNS_DB.plant_readings(device, time, readings) VALUES
    (""+WiFi.macAddress()+""",NOW(),""+jsonstr+"");";

    char query[512];
    q.toCharArray(query, 512);
    cur_mem->execute(query);

    jsonBuffer.clear();

    mvalue1=0;

    hvalue1=0;

    tvalue1=0;

    lvalue1=0;

}

}

```

```
}
```

Arduino MEGA Code

```
#INCLUDE <SOFTWARESERIAL.H>

#include "DHT.H"

#define DHTPIN 9 // DIGITAL PIN CONNECTED TO THE DHT SENSOR

#define DHTTYPE DHT11 // DHT 11

DHT DHT(DHTPIN, DHTTYPE);


CONST INT ARDUINORXPIN = 6; //CONNECT TO BLUESMIRF TX

CONST INT ARDUINOTXPIN = 7; //CONNECT TO BLUESMIRF RX

CHAR CHIN ;

CHAR CHOUT;

CHAR CR = 0x0D;

INT POTPIN = 0; // ANALOG PIN USED TO CONNECT THE POTENTIOMETER

INT VAL; // VARIA

BOOL S1 = TRUE;

BOOL S2, S3, S4 = FALSE;

SOFTWARESERIAL SMIRF(ARDUINORXPIN, ARDUINOTXPIN);


VOID SETUP() {

    SERIAL.BEGIN(9600);

    SERIAL.PRINTLN(F("STARTING BLUESMIRF TESTER." ));


    PINMODE(ARDUINORXPIN, INPUT);

    PINMODE(ARDUINOTXPIN, OUTPUT);


    PINMODE(A0, INPUT);

    PINMODE(A1, INPUT);

    SMIRF.BEGIN(9600);

    SMIRF.LISTEN();

    DHT.BEGIN();

}

VOID LOOP() {

    FLOAT H = DHT.READHUMIDITY();
```



```

// READ TEMPERATURE AS CELSIUS (THE DEFAULT)

FLOAT T = DHT.READTEMPERATURE();

INT M = ANALOGREAD(A0);

INT L = ANALOGREAD(A1);

IF (S1 == TRUE) {

    SERIAL.PRINTLN("M");

    SERIAL.PRINTLN(String(M));

    SMIRF.PRINT('M');

    SMIRF.PRINT(String(M));

    S1 = FALSE;

    S2 = TRUE;

    S3 = FALSE;

    S4 = FALSE;

}

ELSE IF (S2 == TRUE) {

    SERIAL.PRINTLN("H");

    SERIAL.PRINTLN(String(H));

    SMIRF.PRINT('H');

    SMIRF.PRINT(String(H));

    S1 = FALSE;

    S2 = FALSE;

    S3 = TRUE;

    S4 = FALSE;

}

ELSE IF (S3 == TRUE) {

    SERIAL.PRINTLN("T");

    SERIAL.PRINTLN(String(T));

    SMIRF.PRINT('T');

    SMIRF.PRINT(String(T));

    S1 = FALSE;

    S2 = FALSE;

    S3 = FALSE;

    S4 = TRUE;

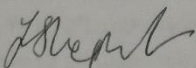
}

```

```
ELSE IF (S4 == TRUE) {  
  
    SERIAL.PRINTLN("L");  
  
    SERIAL.PRINTLN(String(L));  
  
    SMIRF.PRINT('L');  
  
    SMIRF.PRINT(String(L));  
  
    S1 = TRUE;  
  
    S2 = FALSE;  
  
    S3 = FALSE;  
  
    S4 = FALSE;  
  
}  
  
DELAY(1800000);  
}
```

APPENDIX B - DELIVERABLES & REQUIREMENTS

Agreement Form: Project Deliverables

Group Number, Names of Team Members, and Programme	Comp 3: Team DDNS Dorota Marczak Nader Sobhi Scott Allan
Programme specialist's Name	Lynsay Shepherd
The deliverables listed below will be submitted by the team by the due date.	
Part A deliverables	To be agreed by programme specialist and team, for example: <ul style="list-style-type: none">- Physical Device- User Manual- Appropriate Documentation- Source Code- Website Code- Functioning Database
Programme specialist's signature	
Team members' signatures	Dorota Marczak S. Sobhi S. Allan

Agreement Form: Requirements

Group Number:

Team members (print): DOROTA MARCZAK, NADER SOBHI, SCOTT ALLAN, LIAM HARLEY

Project Title: Smart Eco System

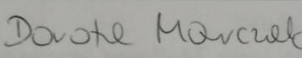
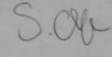
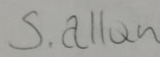
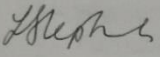
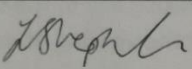
Please refer to the attached documentation for full details on the project. The requirements are listed in Table 1. The signatures below indicate that the requirements for this project have been agreed by the project stakeholders.

Any changes to the project documentation should be made using the correct change authorisation procedure agreed with the program specialist.

ID	List of Agreed Requirements
Functional Requirements:	
F-1.	Wemos device connects to the database
F-2.	Gather data on plants (Temperature, Light, Moisture, Humidity)
F-3.	Put this Data into a database
F-4.	Present the data in a graphical manner on website
F-5.	Feature in place on website to register device
F-6.	Feature in place for users to register
F-7.	Feature in place for users to login
F-8.	Require device-account registration
F-9.	Require registering on the website before utilising any of the services
F-10.	Require login on the website before allowing the viewing of data
F-11.	Present current status on device itself
F-12.	Section of website accessible to all users with information on the Clients company
F-13.	Section of website accessible to all users with information on the functionality of the device
F-14.	Section of website accessible to all users with information on the system as a whole
F-15.	Section of website accessible to all users with contact information of the clients company
F-16.	The website will give users the option to select the time length they want to view the data
F-17.	Email user when status of plant is low in any of the measurements
F-18.	Website must be responsive
F-19.	Users should be able to clear data pertaining to them

Non-Functional Requirements:

- NF-1. Website should be accessible to all users
- NF-2. Website and database should be able to be scalable to user size
- NF-3. Device should be easy to setup for users
- NF-4. Device should be able to be produced on a commercial scale
- NF-5. Website should be user friendly
- NF-6. Device should be reliable in terms of data handling and performance
- NF-7. System as a whole should be available with as little interruption as possible
- NF-8. Website should be easily maintainable after it is delivered to the user
- NF-9. Websites login and register features should be secure to malicious attacks or tampering
- NF-10. Data collected and stored in a manner that is compliant with the relevant laws of the region of operation
- NF-11. The data's integrity should be preserved throughout its life-cycle
- NF-12. Database should be secure and non-accessible to users
- NF-13. Device has some functionality in place to ensure that data loss is minimal
- NF-14. Ensure that the method with which data is collected and held complies with the GDPR
- NF-15. Inform the user of what data will be collected and for what purpose in order to comply with the GDPR

Stakeholders	Signatures	Date
<u>Team members</u> Dorota Marczak Nader Sobhi Scott Allan Liam Harley	  	18/03/19
<u>Programme Specialist</u> Lynsay Shepherd		12/03/19
<u>Client</u> Lynsay Shepherd		12/03/19

APPENDIX C - MINUTES

Meeting One

Date, Time & Venue

15th January, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Nader Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

N/A

Topics of Discussion

The team discussed and formulated the functional and non-functional requirements of the project. These were then paired with the features the project should have in order to ensure the maximum potential of the project could be achieved within the set timescale. These features were then divided into those necessary for the project to be realised and those which are desirable. These were features that, if not developed, would not impact the basic operation and success of the project but were deemed as something that would increase the overall functionality of the device.

Preliminary research into current technologies to meet the requirements set out was also undertaken, with the main focus being on how to turn the sensor data into visually attractive and intuitive graphs.

Challenges

The team was challenged by the sheer amount of options at their disposal. Whilst determining the features that the project should contain was relatively straightforward, the options to realise these features was rather overwhelming and required more in-depth research. Overall, the team were excited to begin development of the project after a relaxing Christmas break.

Actions for Next Meeting & Details

The task for the next meeting was to settle on which technologies to use, as well as draw up paper prototypes for the website.

Meeting to be held 22nd January, 11am, Room 4510, Kydd Building, Abertay University

Meeting Two

Date, Time & Venue

22nd January, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Nader Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The previous meeting consisted of formulating the requirements of the project, turning them into specific features, separating these feature into those that were deemed absolutely necessary and those that were desirable if time was left and a discussion of the technologies available to complete the project.

Topics of Discussion

This week, the team settled on which technologies to use with back up technologies listed in case system integration was unsuccessful or problematic. Paper prototypes of the website had also been developed. The team then split into two sub teams: Dorota would begin development of the website while Nader and Scott began development of the hardware device. Dorota planned to develop three digital prototype of the website to see which features worked best. Nader and Scott began developing the hardware by ensuring that the Arduino and WeMos boards could communicate with each other. Circuit diagrams were drawn up to keep track of wiring connections using Fritzing.

Challenges

There were no real challenges this week as development had just begun and as such it was too early for real problems to arise. The team were confident that the technologies they had chosen were fit for purpose, now it was a case of implementing them correctly.

Actions for Next Meeting & Details

Complete digital prototypes of website, evaluate them, extract the best features and combine them into a final prototype to test usability and responsiveness.

Ensure that the Arduino and WeMos can communicate over a sustained period of time without faults or errors appearing.

Meeting to be held 29th January, 11am, Room 4510, Kydd Building, Abertay University

Meeting Three

Date, Time & Venue

29th January, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The previous meeting consisted of agreeing which technologies to use and drawing up a list of alternatives should they be required. The team then split in two so that work on the website and hardware device could begin concurrently.

Topics of Discussion

This week, the team evaluated the different prototypes Dorota had created and selected the best features from each. These would then be integrated into a final prototype. Also looked at was which sensors to use to collect data on the plant. Scott already had a DHT11 sensor that could detect temperature and humidity levels and it was agreed this was a good starting point to test integration. Sensors to monitor light and moisture level were also selected and purchased.

Dorota began developing the final website prototype, while Scott and Nader integrated the DHT11 sensor with the device and tested whether the data returned was useful.

Challenges

The team spent some time going over the digital prototypes of the website and selecting the best features. However, everyone's idea of 'best' was slightly different, leading to quite a heated discussion. Each feature was then put to a vote, with a feature requiring at least two votes to be selected. This was quite a simple but effective solution.

Scott and Nader had slight difficulties integrating the DHT11 sensor with the Arduino, but upon realising that it required its own library to operate, this problem was resolved.

Actions for Next Meeting & Details

Complete development of functionality of website prototype and integrate all sensors with Arduino. Look at database solutions and how to store data.

Meeting to be held 5th February, 11am, Room 4510, Kydd Building, Abertay University

Meeting Four

Date, Time & Venue

5th February, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The previous meeting consisted of extracting the best features from three separate digital prototypes and integrating them into one prototype.

The hardware device also gained its first sensor and was tested to ensure correct integration.

Topics of Discussion

This weeks meeting was brief, with the team going over what they had achieved at the last meeting before continuing development on the website and hardware device.

Challenges

Dorota reported no challenges from the website development, while Nader and Scott had some difficulties integrating all the peripherals with the chosen Arduino board. The team discussed dropping a component from the hardware device, but it was agreed this would have a major impact on the success of the project and the usefulness of the device. Instead, the team decided to use a slightly more powerful Arduino board which Scott owned and would bring for next time.

Actions for Next Meeting & Details

Continue website development

Scott to bring Arduino Mega 2560 to replace Arduino Uno

Meeting to be held 12th February, 11am, Room 4510, Kydd Building, Abertay University

Meeting Five

Date, Time & Venue

12th February, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The previous meeting minutes were reviewed before the team convened this weeks meeting

Topics of Discussion

The team took some time to consider what they had achieved so far and what still needed to be done. It was found that development of the website was progressing at a steady rate while the hardware side had fallen behind schedule. It was agreed this was due to unforeseen difficulties integrating all the components with an Arduino Uno and, provided the Mega 2560 proved to be a better option, the project could get back on track without much trouble.

Fortunately, this proved to be the case and thanks to the circuit diagrams that had been kept for the Uno, integrating the Mega 2560 and sensors into the system proved rather trivial. The device was tested by returning values to the the Serial Monitor, after which a database was set up in order to retain this data.

Challenges

The main challenge this week was for Nader and Scott to get back on track with the hardware device. Dorota was nearing the point where she was ready to begin developing the graphs that would inform the user of the plants status. Without data to power these, she would have to rely on dummy data that may not be representative of the returned data from the sensors. However, Nader and Scott successfully got the hardware device to a basic operational state and had begun sending recorded data to the database which Dorota could use to power the graphs/charts on the website.

Actions for Next Meeting & Details

Record large amount of information from sensors for website

Identify best format for displaying this data

Reassess how to store the data efficiently in the database

Meeting to be held 19th February, 11am, Room 4510, Kydd Building, Abertay University

Meeting Six

Date, Time & Venue

19th February, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

Upon review of the previous minutes, it was agreed the project was proceeding to schedule after a small bump in the road regarding the Arduino board.

Topics of Discussion

After setting up the database last week, it was decided that the manner in which the data was being stored was not as efficient as it could be. Alternative methods were looked at before settling on trying to append the sensor data into a JSON string which then would be sent to the database, reducing the amount of columns required. This would also make it easier for Dorota to parse the data on the website side.

Dorota was concerned that the technology used to create the graphs was not accurate enough and indicated that she would attempt to integrate different software to power the graphs, hopefully leading to more accurate representations.

Challenges

The challenges facing the team this week were refining the design already in place to make it more efficient and accurate. Initially the team felt that this might set them back, but after discussion and consideration, it was decided that any potential set back was outweighed by the advantage of being able to reflect on aspects of the project that could be improved before development had reached a stage of completion.

Actions for Next Meeting & Details

Dorota to implement different graph visualisation software

Nader and Scott to construct function to create JSON string, append data and send to database

Meeting to be held 26th February, 11am, Room 4510, Kydd Building, Abertay University

Meeting Seven

Date, Time & Venue

26th February, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The team spent some time reflecting on the decisions made last week and agreed that it was worthwhile reviewing the status of the project and make improvements and refining what was already in place.

Topics of Discussion

Dorota felt more comfortable with the new software chosen to power the graphs as the code was easier to work with and provided more fine control on how to visualise the data.

Scott and Nader were having trouble appending all the data to the JSON string without errors occurring. It was clear this needed more work, but the data being sent to the database currently was enough for Dorota to work with in the mean time.

Challenges

The main challenge this week was trying to figure out why the JSON string was only willing to accept certain data. This lead to some frustration for Scott and Nader as the nature of programming Arduino boards means there is no debugging feature to provide assistance. This frustration also bled into Dorota as she wanted more data to work with in order to produce more elaborate graphs. However, during a coffee break this frustration was smoothed over as stepping away from the project allowed the team to conduct some 'out of the box' thinking in order to create ideas and plans to tackle the problem.

Actions for Next Meeting & Details

Solve the JSON string problem

Meeting to be held 5th March, 11am, Room 4510, Kydd Building, Abertay University

Meeting Eight**Date, Time & Venue**

5th March, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

Reviewing the minutes from last week, the team all agreed that stepping away from the project to talk things over led to valuable insights and allowed team members to postulate solutions that, while perhaps not entirely accurate, held the seed of an idea that could be developed into negating the problem. The team agreed that a short break while attempting to solve any future problems would be healthy and lead to a more collaborative and engaging experience for all those involved. The team also agreed they should have come to this conclusion earlier in the project.

Topics of Discussion

The team came to a conclusion of the best way to visualise the data by splitting it into day, week and monthly views. This would provide the users with large amounts of data to formulate informed decisions. Dorota began developing this with the data already recorded, while Scott and Nader tried to solve the JSON string problem using the ideas formulated at the last meeting.

Challenges

The team had to stay positive in order to solve the JSON string problem. Whilst it would be easy to quarrel and get downbeat, each team member had a role to play in motivating the others and ensuring that no one felt overwhelmed or under too much pressure.

This led to the JSON string problem being solved by the end of the meeting, which meant that the core functionality of the project was now complete and had only to be refined, refactored and polished to ensure a professional level result.

Actions for Next Meeting & Details

Review the project as a whole, suggest improvements where necessary and refactor the codebase so that it is simple as possible.

Meeting to be held 12th March, 11am, Room 4510, Kydd Building, Abertay University

Meeting Nine

Date, Time & Venue

12th March, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

After the success of last week, it was important the team kept their feet on the ground. Whilst the project was nearing completion, it was agreed it was important to use the time remaining to improve the project where possible.

Topics of Discussion

The team discussed potential improvements and listed them in order of importance.

Challenges

Dorota stated she was having some issues displaying the data in the manner she wished. Nader agreed to work with her to solve the issue, while Scott began refactoring the code and looked at improvements to the wiring schematic.

Actions for Next Meeting & Details

Refactor current code base

Implement any improvements deemed feasible in the time left

Solve data display issues

Meeting to be held 19th March, 11am, Room 4510, Kydd Building, Abertay University

Meeting Ten

Date, Time & Venue

19th March, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The team reviewed the list of improvements suggested last week and decided to implement a screen that would convey real time data to the user.

Topics of Discussion

The team spent a short time talking over which display to use and what data it should show. This fed into a discussion about graphs on the website, after which it was concluded the website should show as much useful information as possible without being overly obtuse and the display device should show only current data.

Challenges

The challenge facing the team this week was deciding what data should be shown on the website and the display device. After reaching an agreement, Dorota and Nader worked on the website data visualisations while Scott worked on integrating the display peripheral without disrupting the flow of data in the system.

Actions for Next Meeting & Details

Refine data visualisations

Complete integration of display peripheral

Meeting to be held 26th March, 11am, Room 4510, Kydd Building, Abertay University

Meeting Eleven

Date, Time & Venue

26th March, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The team spend sometime reviewing last weeks minutes as well as the project as a whole.

Topics of Discussion

The team discussed how happy they were with the project up until this point. No one raised any concerns and it was agreed that the project had developed well, despite a few challenging weeks over the duration. With the Easter break approaching, the team concluded that the website and device were essentially complete and the break could be used to spend some time individually reflecting on the project as a whole.

Challenges

The team all felt positive regarding the outcome of the project up to this point. The challenge was ensuring that the website and device were as complete as could be expected before the Easter break, after which the team would begin work on the White Paper.

Actions for Next Meeting & Details

Meeting to be held 16th April, 11am, Room 4510, Kydd Building, Abertay University

Meeting Twelve

Date, Time & Venue

16th April, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

The team welcomed each other back after the Easter Break and reviewed the project as a whole.

Topics of Discussion

With the website and device complete, the team discussed the White Paper and divided the work accordingly.

Challenges

With three people writing the White Paper concurrently, it was essential the tone and flow of the report was similar to ensure that it was easy to read and follow. Dorota and Nader raised some concerns that as non-Native English speakers, their sections may suffer. Scott reassured them that their language skills were better than they gave themselves credit for and reminded them that the team would all read over the White Paper together before submission so that anything that seemed lacking could be altered or added.

Actions for Next Meeting & Details

Keep working on White Paper and highlight any areas that may require assistance in completing.

Meeting to be held 23rd April, 11am, Room 4510, Kydd Building, Abertay University

Meeting Thirteen

Date, Time & Venue

23rd April, 11am, Room 4510, Kydd Building, Abertay University

Attended By

Dorota Marczak, Naber Sobhi & Scott Allan

Absences & Apologies

N/A

Review Previous Minutes

Last weeks minutes were looked at before work on finalising the White Paper was undertaken.

Topics of Discussion

With the project submission due at the end of the day, the team reviewed the White Paper and suggested alterations where required.

Challenges

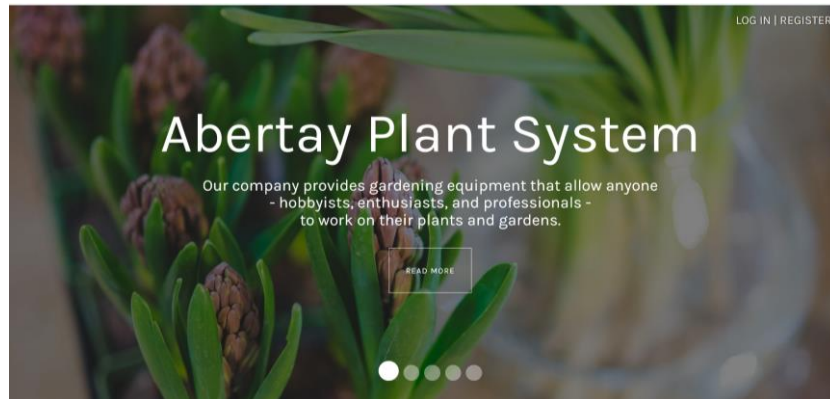
The team faced the challenge of ensuring the length of the White Paper met the word count stipulated. The team agreed not to be overly precious if the section they worked on had to be heavily altered as the result affected each team member equally and not individually.

Actions for Next Meeting & Details

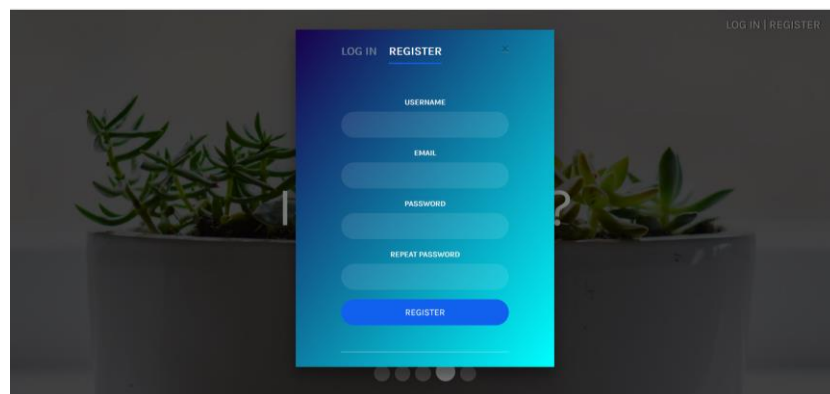
N/A

USER MANUAL

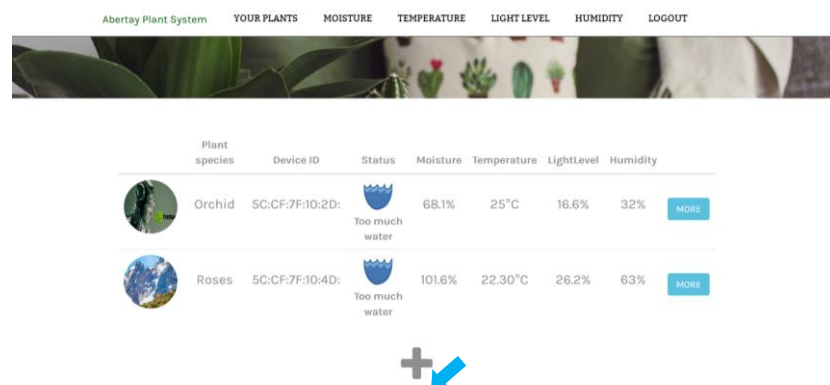
1. Visit the online address - <http://nadersobhi.website/view/index.php>



2. Register an account by clicking on the button "Log in|register" and filling in the form




3. Click "+" to add new plant to your list (follow the blue arrow)



4. Fill in the form adding the image of your plant, key (ID of the device), plant species and environmental preferences, and press the 'submit' button to confirm the operation.

Add another plant to your unique collection ✕



Your plant's image

Nie wybrano pliku.

Key
(please check the number located on device's case)

Plant species

Light demand

☒ Full Shade ☐ Partial Sun/Shade ☐ Full Sun

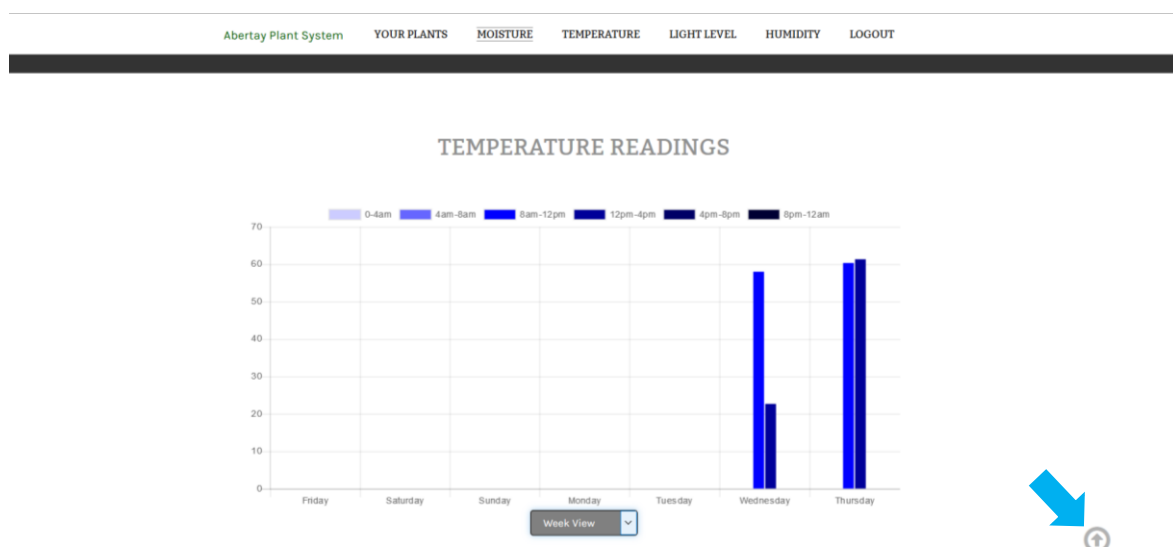
Water demand

☒ Xerophyte (Low) ☐ Mezophyte (Medium) ☐ Hyhrophyte (High)

Environment preferation

☒ Low temperatures ☐ Mederated temperatures ☐ High temperatures

5. Choose the plant by clicking 'more' (step 3) and browse the graphs at your leisure. Move between them even faster with 'scroll-up' button (blue arrow) or use the menu, which will automatically carry you to the desired section.



6. Decide on the time range of the analysed readings using the options (orange arrow)

